IL LINGUAGGIO SEL

MANUALE DI ISTRUZIONI

Prima Edizione

Modelli supportati

Modello	**Formati dei file prodotti**						**Versione Minima**
	Program (Singolo)	Program (Raccolta)	Posizioni	Parametri	Simboli	Dati Globali	
X-SEL-J/K	xpg	xpa	xpt	xpm	xsm	xgd	1.0.0.0
X-SEL-JX/KX	spg	spa	spt	spm	ssm	sgd	2.0.0.0
X-SEL-P/Q	x2pg	x2pa	x2pt	x2pm	x2sm	x2gd	3.0.0.0
TT	tpg	tpa	tpt	tpm	tsm	tgd	4.0.0.0
X-SEL-PX/QX	s2pg	s2pa	s2pt	s2pm	s2sm	s2gd	5.0.0.0
SSEL	sspg	sspa	sspt	sspm	sssm	ssgd	6.0.0.0
ASEL	aspg	aspa	aspt	aspm	assm	asgd	7.0.0.0
PSEL	pspg	pspa	pspt	pspm	pssm	psgd	7.0.0.0

Termini di licenza sull'utilizzo del software

Grazie per l'acquisto di un prodotto IAI.
L'utilizzo di questo software vi è garantito solo in seguito all'accettazione dei termini di licenza. Prima di utilizzare il software leggete con attenzione questo regolamento.
SINTA S.r.l garantisce all'utente, previa accettazione, il diritto non esclusivo e non trasferibile all'utilizzo del programma software fornito con questi termini di licenza.

1. **Accettazione dei termini**
   L'accettazione dei termini di utilizzo incomincia con l'avvio del Software protetto da licenza e rimane valido finchè l'utilizzatore non ne richiede esplicitamente la cancellazione. Alternativamente può terminare in seguito a quanto definito al punto 5.

2. **Diritto di utilizzo del Software**
   L'utente ha diritto ad utilizzare il software su più computer. L'utente non ha il diritto di vendere, trasferire o affittare il diritto di utilizzo del software a terze parti, senza averne richiesto in precedenza l'autorizzazione a SINTA. Senza specifica autorizzazione l'utente non è autorizzato a copiare o riprodurre il software nella sua totalità o anche in parte.

3. **Duplicazione del Software**
   L'utente non è autorizzato a duplicare il software, senza averne richiesto in precedenza l'autorizzazione a SINTA. La realizzazione di un duplicato è consentita solo ai fini della realizzazione di una copia di sicurezza del software.

4. **Protezione del Software**
   L'utente è autorizzato a fornire il software a terze parti interne all'azienda o a tecnici SINTA (IAI).

5. **Termine della licenza**
   Nel caso di infrazione di una della clausole presenti in questa lista, o in seguito alla determinazione di una causa materiale che renda impossibile il mantenimento della licenza, SINTA la può revocare immediatamente senza fornire alcuna precedente notifica all'utilizzatore. In questo caso l'utente deve rimuovere il software ed i suoi duplicati entro dieci (10) giorni, ed inviare una conferma della cancellazione alla SINTA.

6. **Responsabilità**
   SINTA si riserva il diritto di modificare una o tutte le condizioni di utilizzo del software senza preavviso. L'utilizzatore acconsente a non richiedere rimborsi a SINTA in caso di danneggiamenti al proprio sistema in seguito all'installazione del software PC Interface for X-SEL.

Avvertenze

1. Il copyright di questo software è di proprietà della IAI Corporation (IAI).
2. Il software ed il manuale di accompagnamento non possono essere utilizzati o duplicati in parte o nella loro interezza senza l'autorizzazione IAI.
3. Il software ed il manuale possono essere utilizzati solo accettando i termini e le condizioni di licenza per l'utilizzo del software.
4. SINTA S.r.l non si assume nessuna responsabilità per eventuali danni o perdite di dati risultanti dall'utilizzo del software.
5. Si prega di notare che la versione o il numero di edizione di questo manuale non corrisponde alla versione del software.
6. Il contenuto di questo manuale può essere soggetto a modifiche senza preavviso.
7. Il software è predisposto per essere utilizzato in ambiente Windows. Questo manuale è stato redatto presupponendo una conoscienza a livello base del sistema operativo Windows.
8. Il linguaggio SEL è utilizzato da più controller IAI. Questo manuale presenta nel dettaglio gli aspetti comuni a tutti i controller della famiglia SEL, con brevi cenni alle caratteristiche proprie di un singolo modello.

Porta utilizzata	**Sistemi operativi compatibili**
RS-232	Windows 98, Windows Me, Windows 2000, Windows XP
USB	Windows 98SE, Windows Me, Windows 2000, Windows XP

Indice

1 Introduzione **1**
- 1.1 Componenti aggiuntive abbinate al software . . . 2
- 1.2 Requisiti di sistema . . . 2
- 1.3 Guida all'installazione del software . . . 3
  - 1.3.1 L'installazione del software . . . 3
  - 1.3.2 Guida all'installazione dei driver per l'adattatore USB . . . 5
- 1.4 Connessione al controller . . . 9
- 1.5 Avviare il software . . . 12
- 1.6 Impostazioni operative . . . 14
  - 1.6.1 Auto-avvio di un programma secondo parametro . . . 15
  - 1.6.2 Avvio delle operazioni tramite segnale esterno . . . 16
  - 1.6.3 Ripristino delle operazioni . . . 17

2 La struttura dei dati nel controller **19**
- 2.1 Il salvataggio dei dati . . . 20
  - 2.1.1 Sistema dotato di batteria per il backup . . . 20
  - 2.1.2 Sistema privo di batteria per il backup . . . 22
  - 2.1.3 Note e avvertenze . . . 25
- 2.2 I dati nel linguaggio SEL . . . 27
  - 2.2.1 Valori e Simboli . . . 27
  - 2.2.2 Tabella Posizioni . . . 34
  - 2.2.3 Comandi . . . 35

3 Spiegazione dei comandi **37**
- 3.1 Comandi . . . 38
  - 3.1.1 Assegnamento di variabili . . . 38
  - 3.1.2 Operazioni Aritmetiche . . . 40
  - 3.1.3 Operazioni Funzionali . . . 43
  - 3.1.4 Operazioni Logiche . . . 46
  - 3.1.5 Comparazioni . . . 49
  - 3.1.6 Timer . . . 50
  - 3.1.7 Operazioni con I/O e Flag . . . 52
  - 3.1.8 Gestione dei programmi . . . 61
  - 3.1.9 Gestione delle operazioni . . . 64
  - 3.1.10 Gestione delle posizioni . . . 67
  - 3.1.11 Controllo degli attuatori . . . 76

3.1.12 Movimentazione degli attuatori 101
3.1.13 IF strutturato 107
3.1.14 DO strutturato 108
3.1.15 Multi-Branch 109
3.1.16 Acquisizione informazioni sistema 110
3.1.17 Zone 111
3.1.18 Comunicazione 112
3.1.19 Operazioni sulle stringhe 113
3.1.20 Gestione pallet 114
3.1.21 Impostazione pallet 115
3.1.22 Movimenti su pallet 116
3.1.23 Pseudo-Ladder 117
3.1.24 Comandi estesi 118

Capitolo 1

Introduzione

Indice

1.1 Componenti aggiuntive abbinate al software 2
1.2 Requisiti di sistema 2
1.3 Guida all'installazione del software 3
1.3.1 L'installazione del software 3
1.3.2 Guida all'installazione dei driver per l'adattatore USB 5
1.4 Connessione al controller 9
1.5 Avviare il software 12
1.6 Impostazioni operative 14
1.6.1 Auto-avvio di un programma secondo parametro 15
1.6.2 Avvio delle operazioni tramite segnale esterno 16
1.6.3 Ripristino delle operazioni 17

1.1 Componenti aggiuntive abbinate al software

Verificate con cura per sincerarvi della presenza dei seguenti oggetti in abbinamento al software di programmazione per controller SEL.

1. Il CD-ROM abbinato al prodotto, contenente il software ed i manuali di riferimento.
2. I cavi di collegamento del PC al controller SEL. [Esclusi dalla fornitura]
   N.B. I cavi di collegamento possono variare in base al controller utilizzato. Si faccia riferimento allo schema seguente.

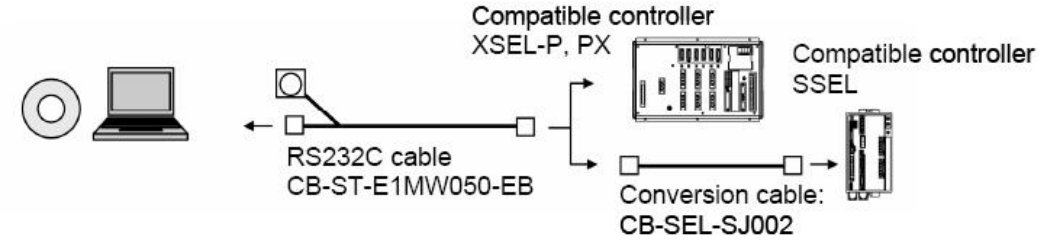


Figura 1.1: Configurazione con collegamento seriale.

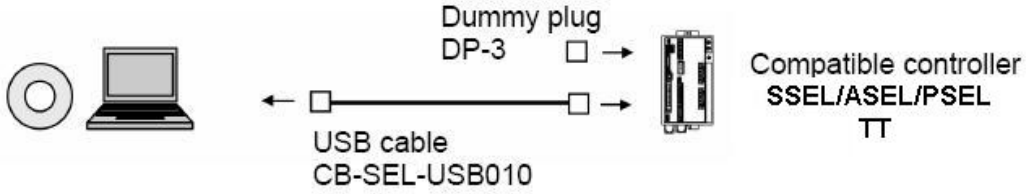


Figura 1.2: Configurazione con collegamento USB.

1.2 Requisiti di sistema

Per il corretto funzionamento del software di programmazione sono richiesti:

1. **PC**
   Un personal computer, dotato di un sistema operativo Windows compatibile con la versione di software in vostro possesso, completo di tastiera e mouse o alternativo sistema di puntamento.
2. **Memoria RAM**
   Il software non richiede un quantitativo minimo di RAM per funzionare correttamente.
3. **Display**
   Monitor VGA o ad alta risoluzione (Si raccomanda un monitor XGA o migliore)
4. **Windows**
   Windows 95, Windows 98, Windows NT, Windows 2000, Windows Me or Windows XP.

5. **Unità CD-Rom**

6. **Porta di comunicazione**
   La comunicazione standard prevede una porta seriale RS-232C (Solo le porte a 9 pin sono supportate). Per alcuni modelli è consentita la comunicazione tramite cavo USB.

1.3 Guida all'installazione del software

Questa sezione illustra la procedura di installazione del software.

1.3.1 L'installazione del software

(Nota) Procedere alla rimozione dell'eventuale versione pre-esistente sul pc (versione 6.0.0.0 o precedente) *prima di installare la più aggiornata.*

---

1. Inserire il CD-ROM contente il software nel drive del PC.

2. Accedere al CD-ROM tramite esplora risorse o programma alternativo. Il software è contenuto nella cartella X:/SOFTWARE - PUBLIC/SOFTWARE SEL dove X corrisponde all'unità CD-ROM del PC.

3. Copiare la cartella SOFTWARE SEL sul computer. L'installazione non è possibile da CD-ROM.

4. Avviare il file Setup.exe e proseguire nell'installazione del software. Comparirà la finestra presentata nella figura 1.3. Selezionare *next*.

PC Interface Software for X-SEL - InstallShield Wizard
Welcome to the InstallShield Wizard for PC Interface Software for X-SEL
The InstallShield(R) Wizard will allow you to modify, repair, or remove PC Interface Software for X-SEL. To continue, click Next.
< Back
Next >
Cancel

Figura 1.3: Avvio dell'installazione.

5. Si aprirà la finestra di registrazione informazioni dell'utente, come presentato nella figura seguente. Selezionare *next* per proseguire.

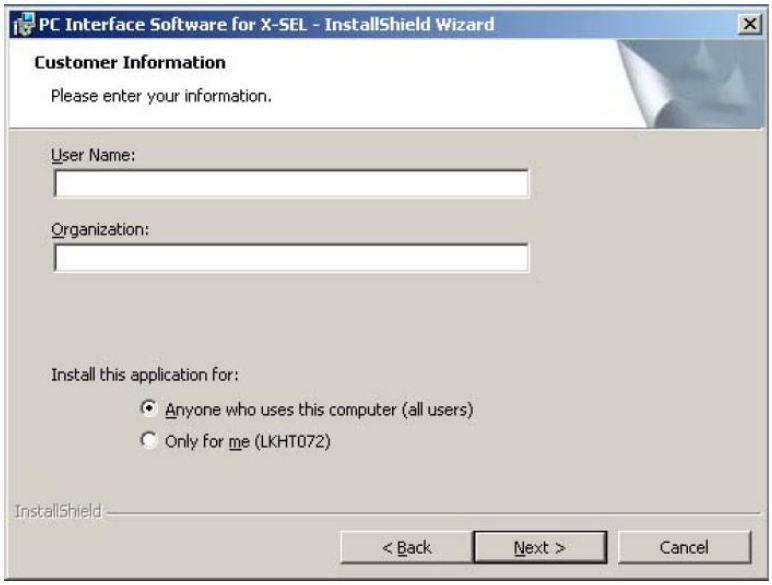


Figura 1.4: Registrazione delle informazioni personali.

6. La finestra successiva consente di impostare la directory di destinazione dell'installazione. La modifica di quest'impostazione è consigliata agli utenti esperti. Selezionare *next* per proseguire.

PC Interface Software for X-SEL - InstallShield Wizard
Destination Folder
Click Next to install to this folder, or click Change to install to a different folder.
Install PC Interface Software for X-SEL to:
K:\Program Files\IAI Corporation\X-SEL\
Change...
InstallShield
< Back
Next >
Cancel

Figura 1.5: Scelta della directory di installazione.

7. Comparirà una finestra riassuntiva contenente un riepilogo delle impostazioni scelte, come in figura 1.6. Selezionare *Install* per proseguire ed iniziare l'effettiva installazione.

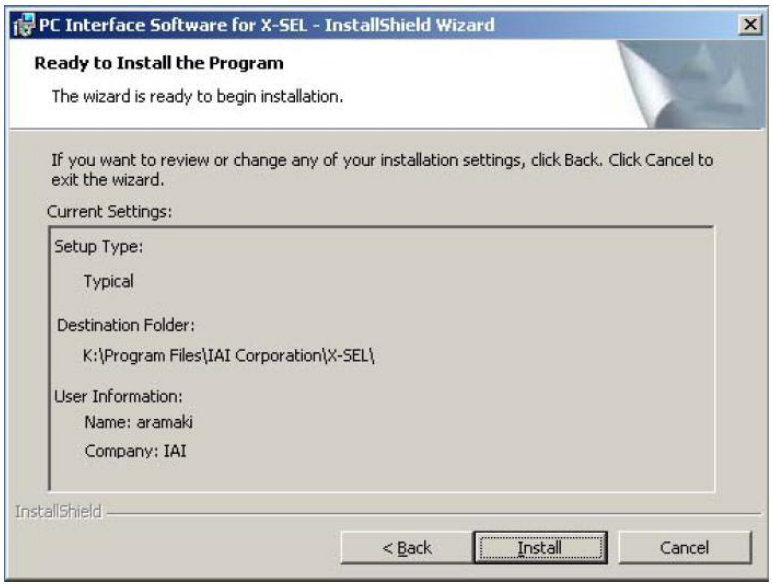


Figura 1.6: Finestra riassuntiva impostazioni di installazione.

8. Al termine dell'operazione comparirà una finestra di conferma dell'avvenuta installazione, e sarà possibile avviare il software.

9. E' ora possibile rimuovere il CD-ROM

1.3.2 Guida all'installazione dei driver per l'adattatore USB

Per poter dialogare con il controller attraverso la porta USB è necessario installare il driver per l'adattatore software da seriale a USB.

(Nota) Il driver USB IAI Port è valido sia per la comunicazione con i Robocilindri che con gli attuatori della famiglia SEL.

---

1. Inserire il CD-ROM contenente il software nell'unità disco.

2. Accedere al CD-ROM tramite esplora risorse o programma alternativo. I driver sono contenuti nella cartella X:/SOFTWARE - PUBLIC/DRIVER USB-485 dove X corrisponde all'unità CD-ROM del PC.

3. Copiare la cartella DRIVER USB-485 sul computer. L'installazione non è possibile da CD-ROM.

4. Avviare il file PreInstaller.exe per l'avvio della procedura di installazione; si aprirà la finestra riportata nella figura 1.7. Impostare liberamente la directory di lavoro temporanea (pulsante Browse) e avviare l'installazione (pulsante Install).

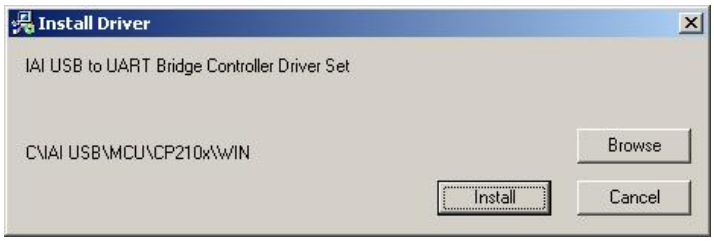


Figura 1.7: Finestra di impostazione della pre-installazione.

5. Al termine dell'installazione comparirà una conferma della buona riuscita dell'operazione.

Figura 1.8: Finestra di buon esito della pre-installazione.

6. E' ora possibile rimuovere il CD-ROM, per procedere al collegamento via USB del controller.
7. Windows aprirà la finestra di riconoscimento nuovo hardware. Proseguire nella procedura.
8. Windows aprirà la finestra di installazione nuovo hardware, proposta in figura 1.9. Selezionare Ricerca di un driver adatto per il nuovo device [raccomandato].

Upgrade Device Driver Wizard
Install Hardware Device Drivers
A device driver is a software program that enables a hardware device to work with an operating system.
This wizard upgrades drivers for the following hardware device:
USB Device
Upgrading to a newer version of a device driver may add functionality to or improve the performance of this device.
What do you want the wizard to do?
Search for a suitable driver for my device (recommended)
Display a list of the known drivers for this device so that I can choose a specific driver
< Back
Next >
Cancel

Figura 1.9: Installazione dei driver.

9. Si aprirà la finestra di ricerca dei file del driver, proposta in figura 1.10. Spuntare l'opzione Specificare una destinazione.

Upgrade Device Driver Wizard
Locate Driver Files
Where do you want Windows to search for driver files?
Search for driver files for the following hardware device:
USB Device
The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.
To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.
Optional search locations:
Floppy disk drives
CD-ROM drives
Specify a location
Microsoft Windows Update
< Back
Next >
Cancel

Figura 1.10: Finestra di localizzazione dei file di driver.

10. Nella finestra di installazione selezionare il pulsante Browse e impostare la directory definita in precedenza nella fase di installazione.

11. Si aprirà la finestra di Risultati della ricerca Driver. Proseguire nella procedura di installazione del driver **IAI USB Composite Device**. Al termine comparirà la finestra di conferma dell'avvenuta installazione driver, come riportato in figura 1.11.

Found New Hardware Wizard
Completing the Found New Hardware Wizard
IAI USB Composite Device
Windows has finished installing the software for this device.
To close this wizard, click Finish.
< Back
Finish
Cancel

Figura 1.11: Fine installazione driver **IAI USB Composite Device**.

12. In seguito si aprirà nuovamente la finestra di Riconoscimento nuovo Hardware. Ripetere la stessa procedura per il **CP2102 USB to UART Bridge Controller**.

13. La procedura di installazione dei driver USB è ora completa.

14. Selezionare il pulsante di **Start** sul menù di avvio di Windows, e quindi aprire Pannello di controllo nel menù Impostazioni.
    Doppio click sull'icona **Sistema** per aprire le Proprietà del sistema.
    Scegliere la slide **Hardware**. Selezionare **Gestione Periferiche**.
    Doppio click sulla lista **Porte (COM e LPT)** per espandere il menù ad albero.
    Se compare **IAI USB to UART Bridge Controller (COM?)** allora l'installazione è andata a buon fine.

(N.B.) Il numero indicato dopo COM, come si può riscontrare in figura 1.12 rappresenta il valore da impostare nel software per ottenere la comunicazione con il controller.

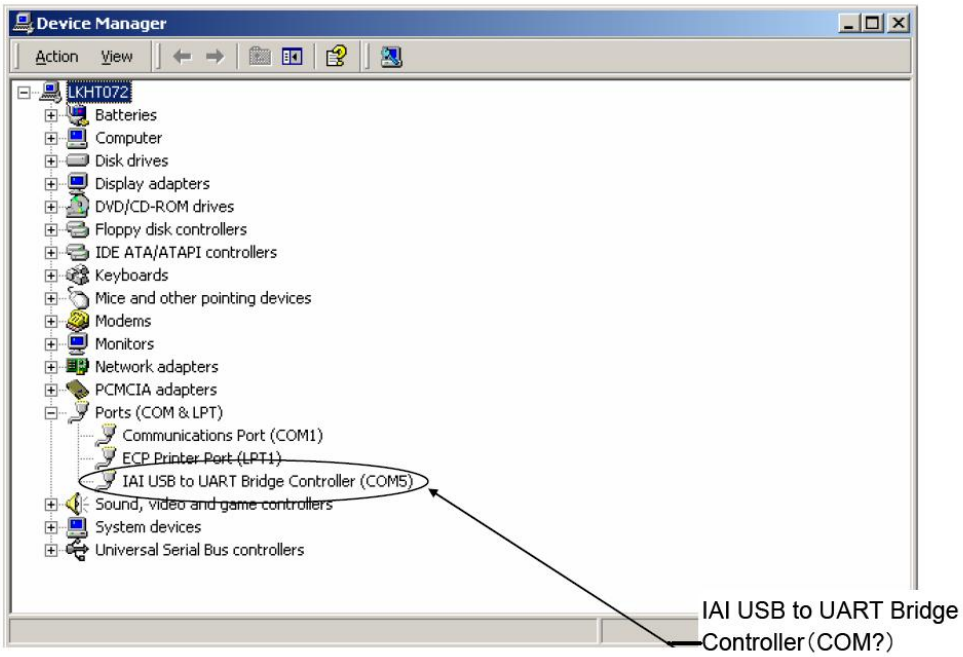


Figura 1.12: Riscontro esito installazione da Gestione periferiche.

1.4 Connessione al controller

Il collegamento ai controller della categoria X-SEL è realizzato sulla porta di comunicazione a 25 pin TP (Teaching Port) situata sul lato frontale del controller, come rappresentato in figura 1.13.

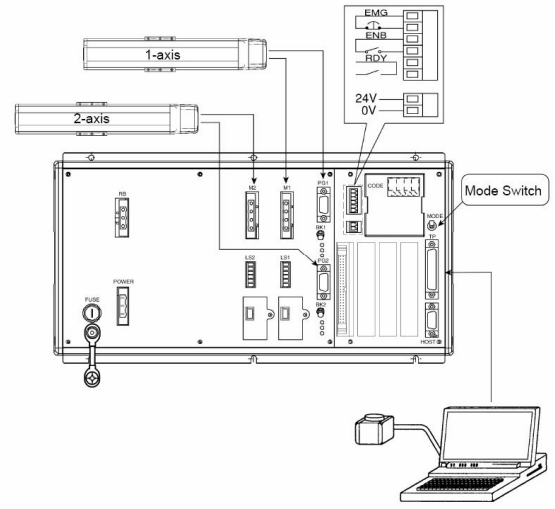


Figura 1.13: Controller X-SEL K(J).

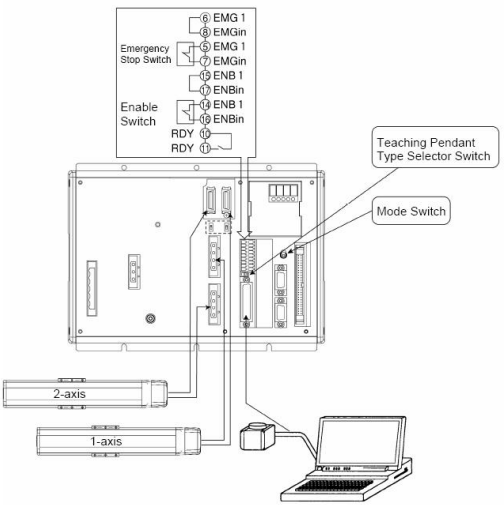


Figura 1.14: Controller X-SEL P(Q).

Il selettore, denominato Mode Switch, posizionato appena sopra alla porta stessa ne determina la modalità di funzionamento:

AUTOMATICO la porta di comunicazione è abilitata per la sola lettura dello stato del controller e delle variabili;

MANUALE la porta di comunicazione è abilitata sia alla lettura che alla scrittura delle variabili (parametri, programmi, punti ...).

I controller TT presentano sia la porta TP di comunicazione seriale, nel formato a 25 pin, sia la porta USB, come rappresentato in figura 1.15.

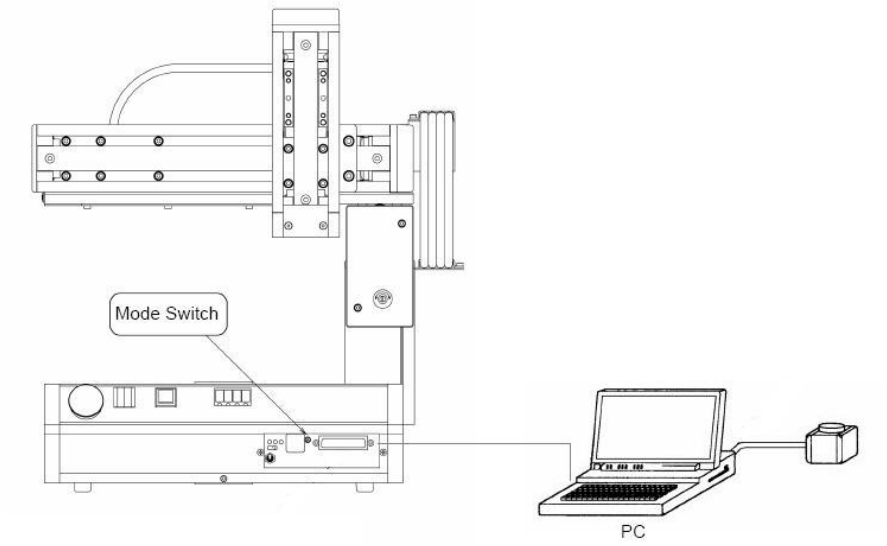


Figura 1.15: Controller TT.

I controller delle categorie SSEL, ASEL, PSEL presentano sia la porta TP di comunicazione seriale, nel formato ridotto che necessita quindi di un adattatore per avere una 25 pin, sia la porta USB, , come rappresentato in figura 1.16.

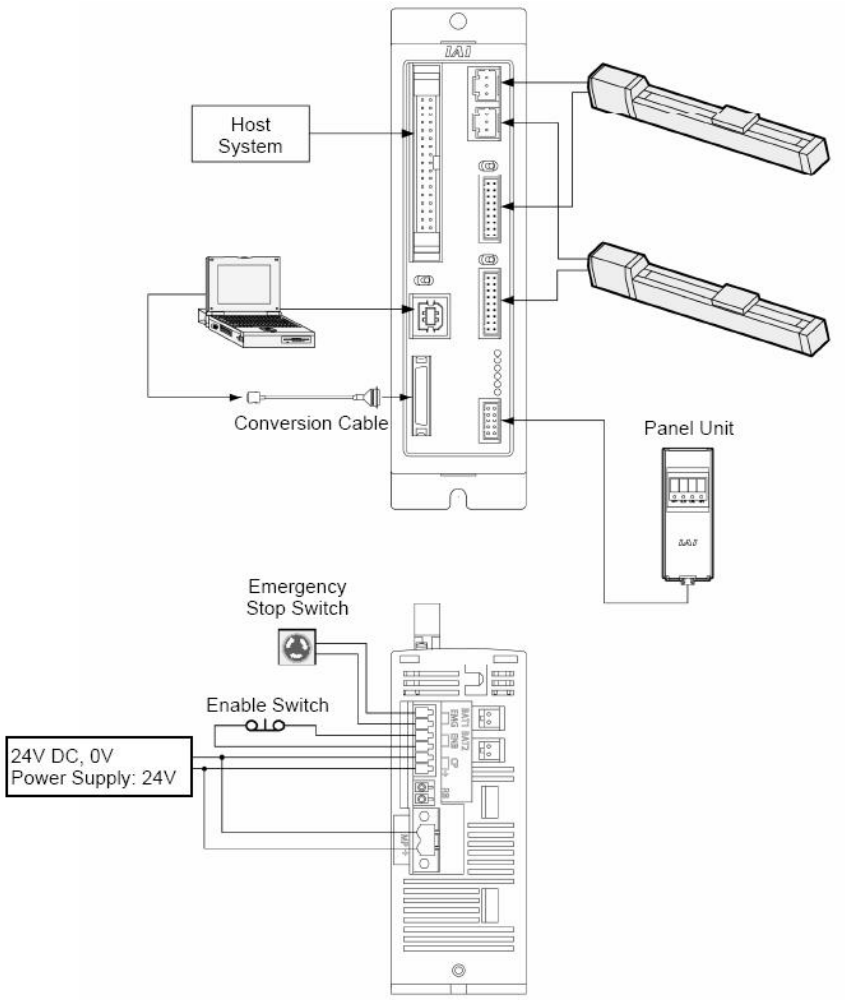


Figura 1.16: Controller ASEL/PSEL.

1.5 Avviare il software

Spegnere il controller ed il PC e collegare il cavo di comunicazione seriale standard RS232C o il cavo USB. Ricordarsi di impostare il selettore frontale nella posizione MANU.

(Nota) All'avvio del software viene abilitato in automatico il limite di sicurezza sulla velocità, pari a 250 mm/s o inferiore, sia per i robot SCARA *che per i sistemi cartesiani. Per testare i programmi a piena velocità si deve disabilitare la modalità* Safety Speed.

Figura 1.17: Impostazioni di apertura porta.

Accendere il controller ed il computer, quindi avviare il software **PC Interface Software for X-SEL**. La prima finestra ad aprirsi consente di definire le impostazioni di connessione, come riportato in figura 1.17, tramite due menù a selezione multipla. Il protocollo di comunicazione non va reimpostato, restano quindi da definire solamente la **Porta COM** su cui effettuare la ricerca dispositivi e la velocità di comunicazione, o **Baud Rate**. Solo le porte di comunicazione libere e accessibili possono essere selezionate quando viene avviato il software. Le velocità marcate con un asterisco sono fruibili solo dai controller X-SEL P/Q, SSEL, ASEL, PSEL. Collegandosi ad un controller di ultima generazione si aprirà la finestra di abilitazione dell'esecuzione contemporanea di due o più programmi, come in figura 1.18

[**Abilitazione all'esecuzione di due o più programmi**]
Consente l'avvio di più di un programma nella modalità manuale. Identificato dall'impostazione *Two or more programs start permission.*

[**Proibizione dell'esecuzione di due o più programmi**]
Proibisce l'avvio simultazione di due o più programmi nella modalità manuale. Identificato dall'impostazione *Two or more programs start prohibition.*

Setting of Two or more programs start
Two or more programs start(MANU)
Two or more programs start permission (MANU)
Don't Show this window
from next time on.
OK
CANCEL

Figura 1.18: Finestra di abilitazione.

Se si spunta la casella *Don't show this window from the next time on*, all'avvio del software verrà stabilita una connessione utilizzando le impostazioni definite nell'ultima sessione di lavoro. Per visualizzare nuovamente la finestra di scelta all'avvio del software accedere alle impostazioni *Environment setup* nel menù *Tool*.

Anche in fase di debug è possibile passare da una modalità all'altra. Il tentativo di cambiare modalità quando più programmi sono in esecuzione produrrà il seguente messaggio di allarme. Selezionando *Yes* verrà interrotta l'esecuzione di tutti i programmi.

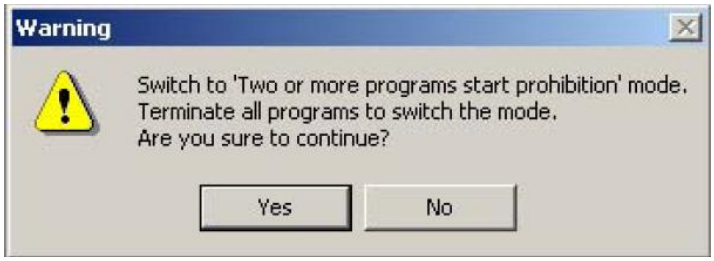

Warning
Switch to 'Two or more programs start prohibition' mode.
Terminate all programs to switch the mode.
Are you sure to continue?
Yes
No

Figura 1.19: Messaggio di allarme.

Il software si avvierà in modalità online solo se la connessione con il controller sarà attiva. Se il riconoscimento del controller dovesse fallire il software partirebbe in modalità offline. E' comunque possibile cercare un controller anche avendo avviato il software in modalità offline, tramite il comando *Reconnect* nel menù Controller.

1.6 Impostazioni operative

Come avviare un programma

I controller della famiglia SEL hanno quattro possibili modalità di avvio di un programma. Due di queste modalità vengono utilizzate principalmente in fase di verifica della programmazione svolta o per effettuare cicli di prova, mentre le rimanenti sono applicate in fase di esecuzione del ciclo di lavoro.

Sia l'avvio tramite tastiera *Teaching Pendant* che tramite il software *PC Interface for SEL* appartengono alla prima categoria. Questi due approcci sono mirati per la fase di messa in funzione della macchina, e mettono a disposizione del programmatore semplici strumenti volti al debug. Si faccia riferimento rispettivamente al manuale della Teaching Pendant(EN) e del software per una trattazione approfondita.

Per la gestione a regime è possibile sia preventivare l'avvio del software con un segnale esterno di selezione, che l'avvio automatico secondo l'impostazione di alcuni parametri.

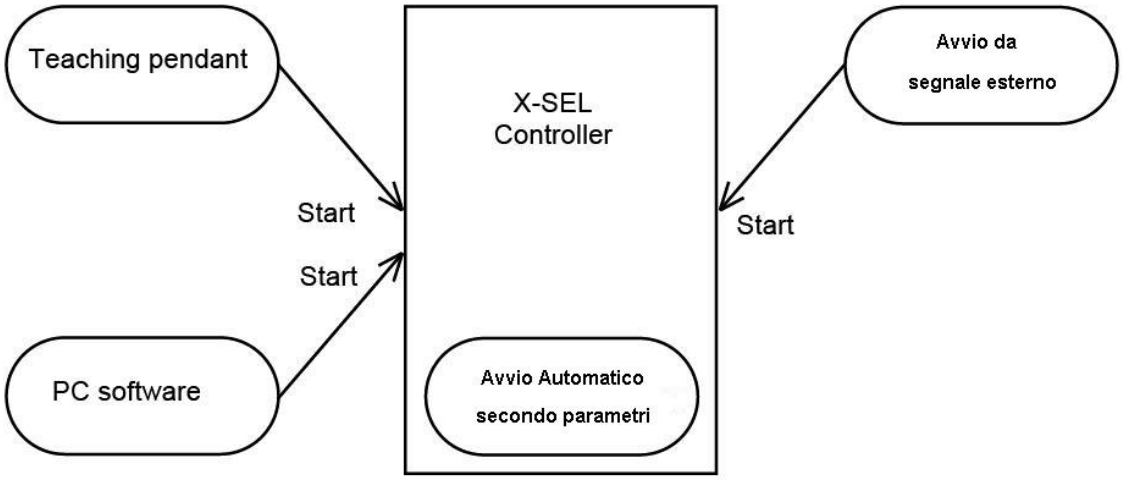


Figura 1.20: Modalità di avvio dei programmi.

1.6.1 Auto-avvio di un programma secondo parametro

(Nota) Per modificare i parametri utilizzare la Teaching Pendant *o il software* PC Interface for SEL.

[Impostazione del programma per l'avvio automatico]

Nella lista degli *Other Parameters* modificare il parametro No. 1 e salvare la modifica nella Flash ROM. Selezionare la modalità AUTO.

[Riavviare il controller]

Spegnere e riaccendere il controller, lasciandolo spento per almeno 20 secondi.

[Avvio automatico del programma]

Terminata la fase di ripristino nella riaccensione il programma impostata verrà avviato in automatico.

Caution

(Note aggiuntive sull'avvio automatico)

L'esecuzione automatica del programma partirà al termine del riavvio del controller. Per garantire un maggior livello di sicurezza è possibile introdurre un ulteriore vincolo quale conferma dell'avvio effettivo del programma[1].

Volendo attivare più programmi allo stesso tempo al riavvio utilizzare il comando EXPG tra le prime istruzioni del programma principale. Ricordarsi di introdurre misure di sicurezza per ogni programma avviato.

[1] Modificando il parametro No. 33 nella lista I/O Parameters, impostando il valore 2, l'avvio automatico del programma avverrà al fronte positivo dell'ingresso digitale No. 3. Un fronte negativo terminerà immediatamente il programma.

1.6.2 Avvio delle operazioni tramite segnale esterno

Il programma[2] da avviare è definito tramite una codifica BCD ottenuta assegnando al controller determinati valori a ingressi predefiniti. L'avvio è realizzato con un segnale di start. Il diagramma di flusso 1.21 fornisce una rappresentazione schematica della sequenza di funzionamento ottenibile.

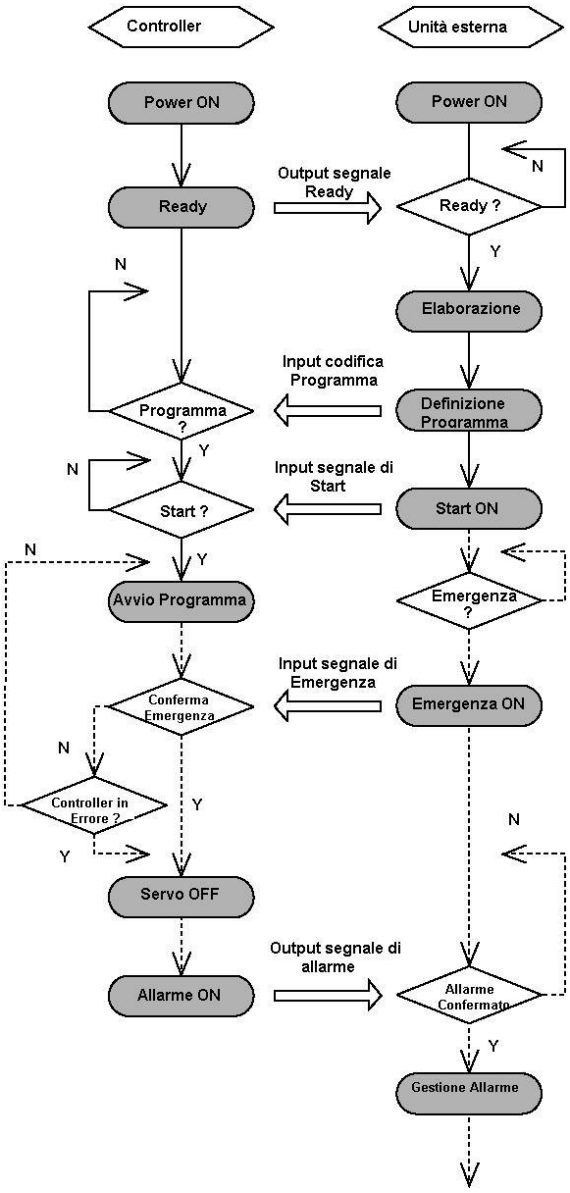


Figura 1.21: Modalità di avvio dei programmi.

[2]I controller X-SEL di ultima generazione, dotati della funzionalità gateway RC, possono memorizzare fino a 128 programmi. Si noti comunque che l'avvio tramite selezione BCD è consentito solo per i programmi compresi tra 1 e 79. Per gli altri si utilizzi il comando *EXPG*.

1.6.3 Ripristino delle operazioni

Richiesta di ripristino del drive di potenza

1. Come richiedere il ripristino
   La richiesta di ripristino del drive di potenza può avvenire in una delle tre modalità seguenti:

   - Impostare il parametro No. 44 degli I/O Parameters a 1, per assegnare all'ingresso No. 14 la funzione di reset. Un fronte positivo su quell'ingresso verrà interpretato come segnale di reset.
   - Selezionare la voce [Drive Source Recovery Request] nel menù [Controller] nell'ambiente software PC Interface for SEL.
   - Selezionare CTL e RPwr nella schermata Mode della Teaching Pendant.

2. Casi in cui è necessario il ripristino
   La richiesta di ripristino deve essere effettuata quando:

   - il parametro I/O No. 44 è stato impostato a 1 e si verifica un evento di blocco del drive di potenza.

Richiesta di ripristino dalla pausa

1. Come richiedere il ripristino dalla pausa
   Una richiesta di ripristino dalla pausa può avvenire in una delle tre modalità seguenti:

   - Impostare il parametro No. 35 degli I/O Parameters a 1, per assegnare all'ingresso No. 5 la funzione di reset. Un fronte positivo su quell'ingresso verrà interpretato come segnale di reset.
   - Selezionare la voce [Operation Pause Reset Request] nel menù [Controller] nell'ambiente software PC Interface for SEL.
   - Selezionare CTL e RAct nella schermata Mode della Teaching Pendant.

2. Casi in cui è necessario il ripristino Il ripristino della pausa è necessario quando:

   - Interviene l'emergenza durante un ciclo automatico, ed il parametro No. 10 degli Other Parameters è stato precedentemente impostato a 2.
   - Il ciclo automatico è stato interrotto con il pulsante uomo-morto o il pulsante di enable, ed il parametro No. 11 degli Other Parameters era stato impostato a 2.
   - Il controller riceve un segnale di OFF sull'ingresso No. 6 quando il parametro No. 36 degli I/O era stato impostato a 1.

Capitolo 2

La struttura dei dati nel controller

Indice

2.1 Il salvataggio dei dati 20
2.1.1 Sistema dotato di batteria per il backup 20
2.1.2 Sistema privo di batteria per il backup 22
2.1.3 Note e avvertenze 25
2.2 I dati nel linguaggio SEL 27
2.2.1 Valori e Simboli 27
2.2.2 Tabella Posizioni 34
2.2.3 Comandi 35

I dati contenuti nel controller comprendono sia i parametri che le posizioni di lavoro memorizzate che i programmi per la gestione dell'applicazione implementati nel linguaggio SEL. L'utente deve creare sia i dati relativi alle posizioni, sia i programmi operativi. I parametri sono invece predefiniti e preconfigurati; sono comunque parzialmente accessibili, per poter essere modificati in base alle necessità dell'utilizzatore.

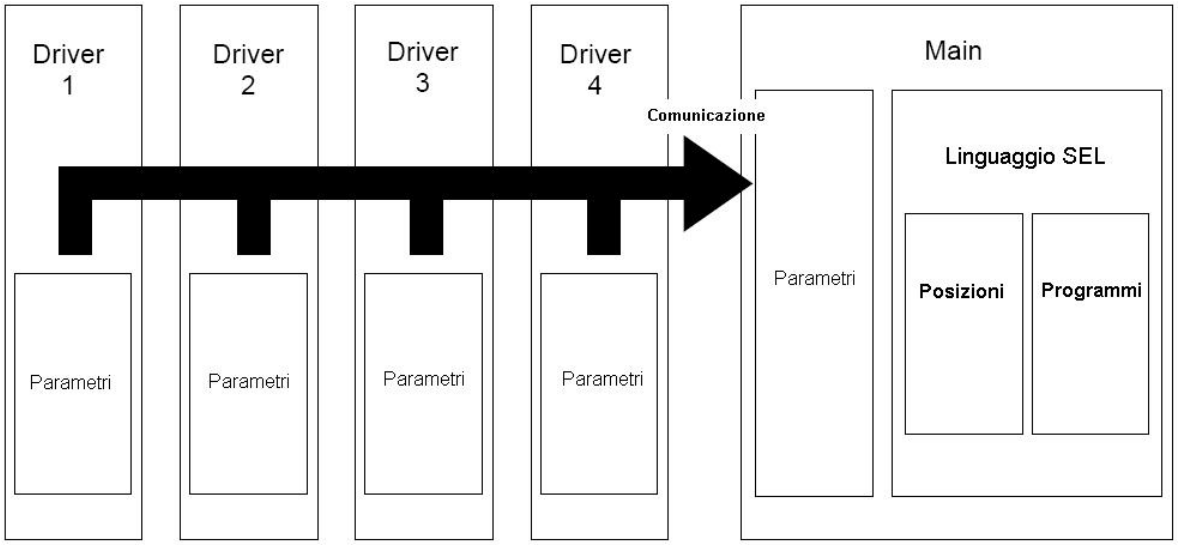


Figura 2.1: La struttura dei dati nei controller SEL.

2.1 Il salvataggio dei dati

I controller SEL sono dotati di una memoria interna della categoria Flash ROM, sulla quale vengono salvati i dati di funzionamento della macchina. Per alcuni dati vi è inoltre la possibilità di realizzare un backup appoggiandosi ad una batteria aggiuntiva che ne consenta la memorizzazione. Il trasferimento dei dati dal PC o dalla tastiera di programmazione al controller immagazzina le informazioni nella CPU del controller, come rappresentato in figura 2.2 ed in figura 2.3. Questi dati verranno cancellati allo spegnimento o al riavvio del controller. Il parametro No. 20 della categoria *Other Parameters* è associato alla gestione della batteria di backup.

(Nota) Ricordasi sempre di trasferire i dati sulla Flash ROM per non perdere il lavoro effettuato.

2.1.1 Sistema dotato di batteria per il backup

Il parametro No. 20 della lista Other Parameters è impostato al valore 2, ad indicare la presenza di una batteria di backup installata.

Controller privo di espansione di memoria

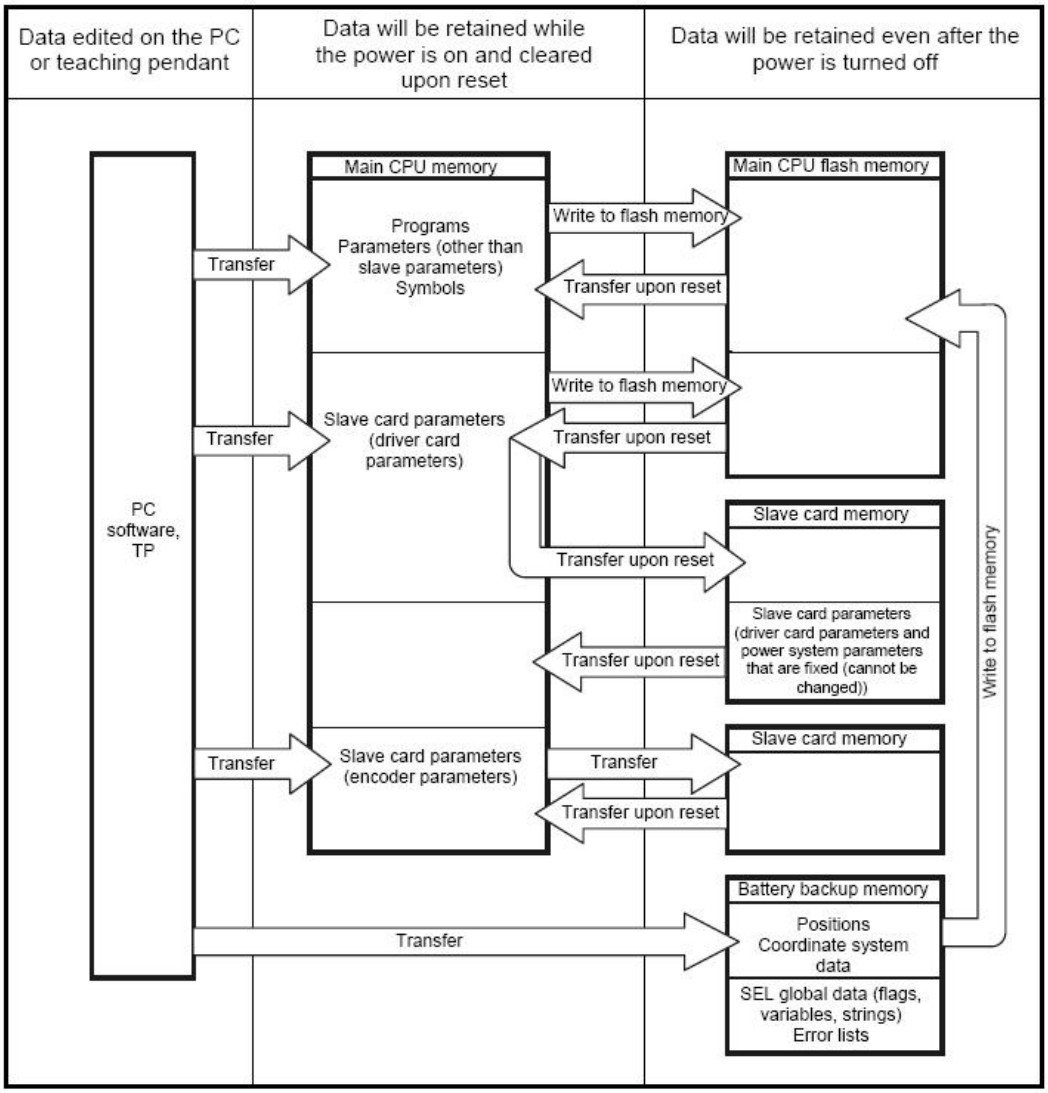

Data edited on the PC or teaching pendant
Data will be retained while the power is on and cleared upon reset
Data will be retained even after the power is turned off
PC software, TP
Main CPU memory
Programs
Parameters (other than slave parameters)
Symbols
Slave card parameters (driver card parameters)
Slave card parameters (encoder parameters)
Transfer
Transfer
Transfer
Transfer
Write to flash memory
Transfer upon reset
Write to flash memory
Transfer upon reset
Transfer upon reset
Transfer upon reset
Transfer
Transfer upon reset
Main CPU flash memory
Slave card memory
Slave card parameters (driver card parameters and power system parameters that are fixed (cannot be changed))
Slave card memory
Battery backup memory
Positions
Coordinate system data
SEL global data (flags, variables, strings)
Error lists
Write to flash memory

Figura 2.2: Il salvataggio dei dati nei controller SEL.

Poichè i programmi, i parametri ed i simboli vengono importati dalla Flash ROM al riavvio, i dati nella memoria temporanea torneranno ad essere uguali ai dati precedenti, a meno di averli salvati nella memoria Flash. Il controller opera sempre conformemente ai dati contenuti nella CPU principale, fatta eccezione dei parametri.

Controller dotato di espansione di memoria (Gateway RC)

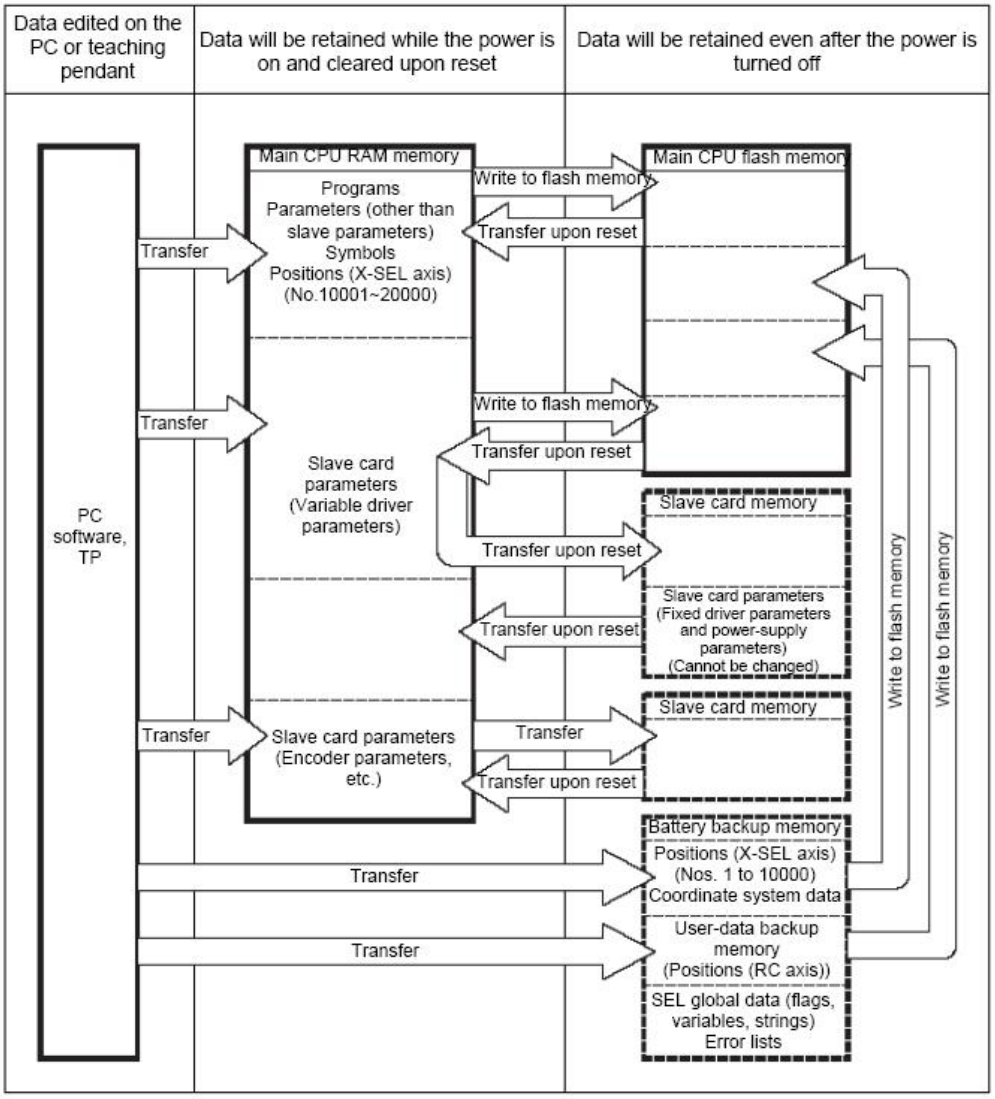

Data edited on the PC or teaching pendant
Data will be retained while the power is on and cleared upon reset
Data will be retained even after the power is turned off
PC software, TP
Main CPU RAM memory
Programs
Parameters (other than slave parameters)
Symbols
Positions (X-SEL axis)
(No.10001~20000)
Slave card parameters (Variable driver parameters)
Slave card parameters (Encoder parameters, etc.)
Main CPU flash memory
Write to flash memory
Transfer upon reset
Transfer
Slave card memory
Slave card parameters (Fixed driver parameters and power-supply parameters)
(Cannot be changed)
Battery backup memory
Positions (X-SEL axis)
(Nos. 1 to 10000)
Coordinate system data
User-data backup memory
(Positions (RC axis))
SEL global data (flags, variables, strings)
Error lists
Write to flash memory

Figura 2.3: Il salvataggio dei dati nei controller SEL.

Poichè i programmi, i parametri ed i simboli vengono importati dalla Flash ROM al riavvio, i dati nella memoria temporanea torneranno ad essere uguali ai dati precedenti, a meno di averli salvati nella memoria Flash. Il controller opera sempre conformemente ai dati contenuti nella CPU principale, fatta eccezione dei parametri.

2.1.2 Sistema privo di batteria per il backup

Il parametro No. 20 della lista Other Parameters è impostato al valore 0, ad indicare la presenza di una batteria di backup installata.

Controller privo di espansione di memoria

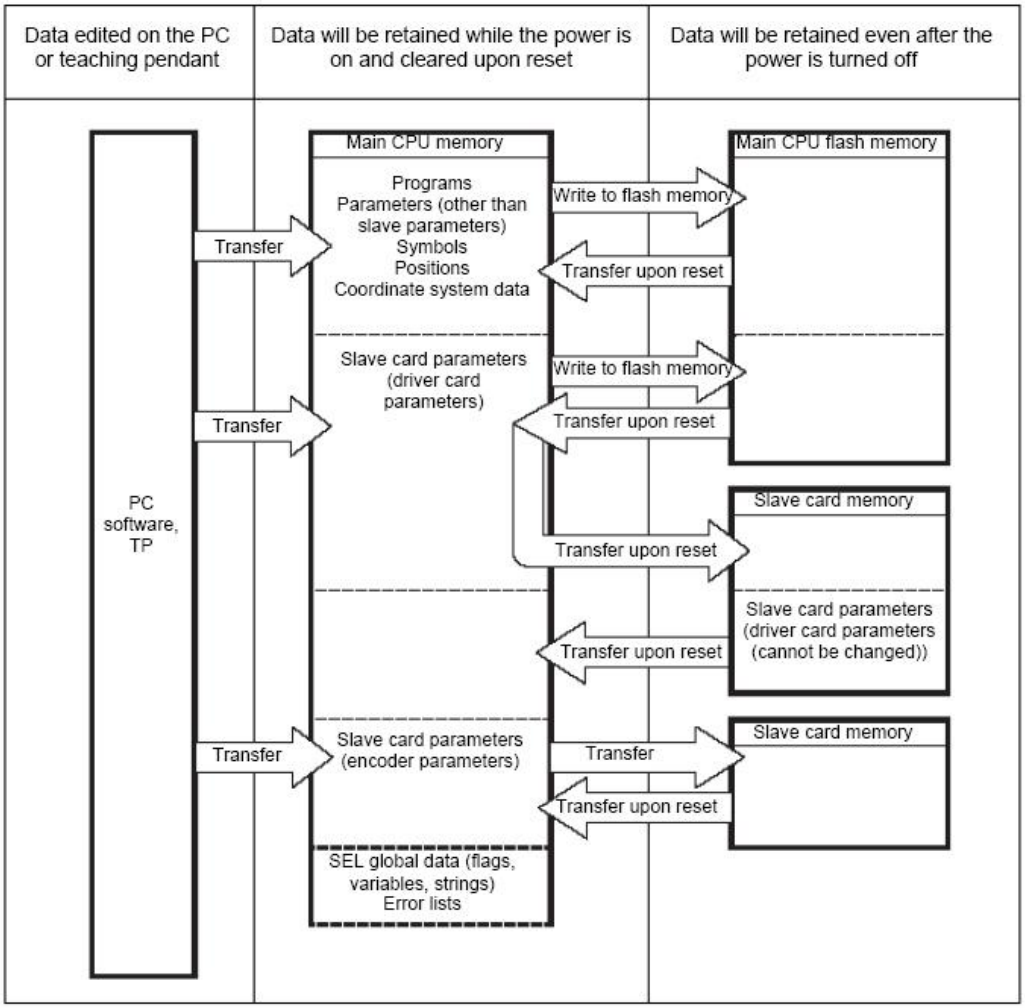


Figura 2.4: Il salvataggio dei dati nei controller SEL.

Poichè i programmi, i parametri ed i simboli vengono importati dalla Flash ROM al riavvio, i dati nella memoria temporanea torneranno ad essere uguali ai dati precedenti, a meno di averli salvati nella memoria Flash. Il controller opera sempre conformemente ai dati contenuti nella CPU principale, fatta eccezione dei parametri.

(Nota) I dati globali non possono essere trattenuti in memoria se non è stata installata la batteria di backup.

Controller dotato di espansione di memoria (Gateway RC)

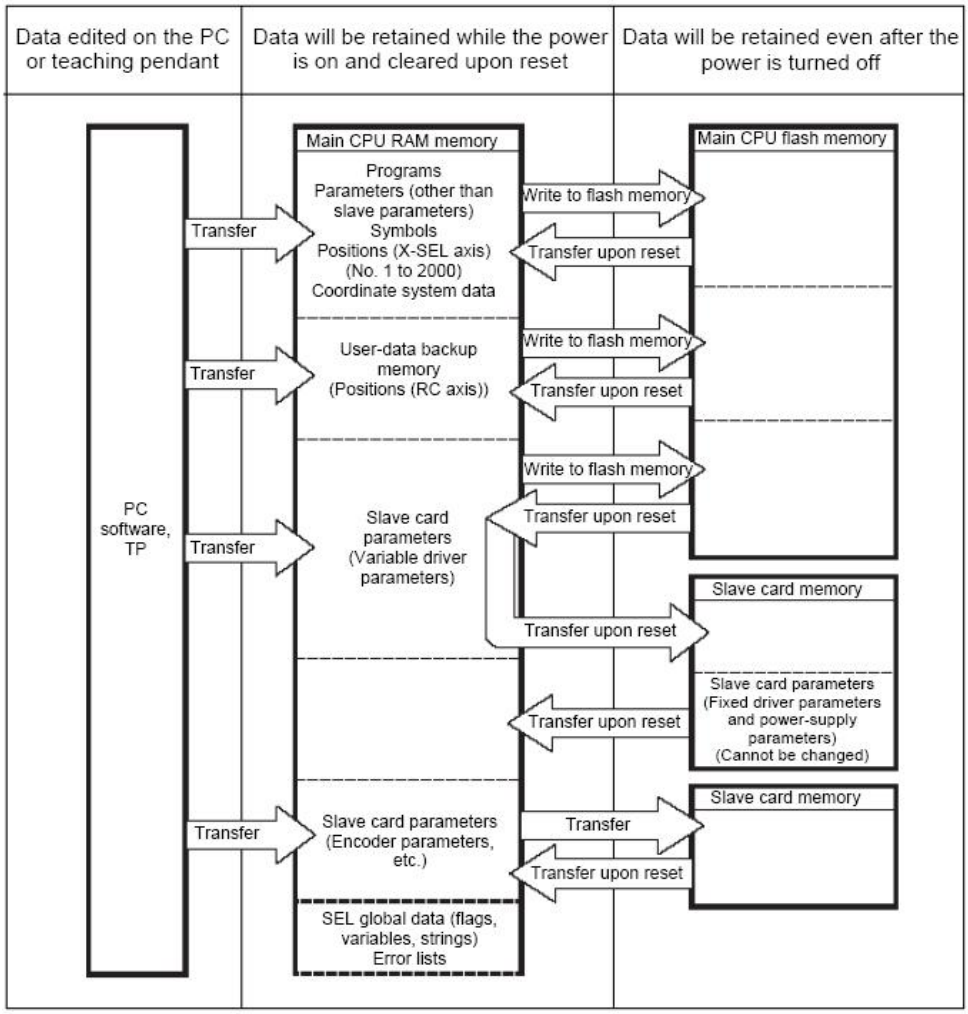


Figura 2.5: Il salvataggio dei dati nei controller SEL.

Poichè i programmi, i parametri ed i simboli vengono importati dalla Flash ROM al riavvio, i dati nella memoria temporanea torneranno ad essere uguali ai dati precedenti, a meno di averli salvati nella memoria Flash. Il controller opera sempre conformemente ai dati contenuti nella CPU principale, fatta eccezione dei parametri.

(Nota) I dati globali non possono essere trattenuti in memoria se non è stata installata la batteria di backup.

2.1.3 Note e avvertenze

Salvataggio sulla Flash ROM

Si faccia attenzione a non spegnere mai il controller durante il trasferimento delle impostazioni o la scrittura dei dati sulla memoria Flash. I dati trasmessi andranno persi, con il rischio di compromettere in modo irreparabile il controller.

Salvataggio di un file dei parametri

I parametri dell'encoder sono memorizzati in una memoria EEPROM collegata allo stesso encoder. Al riavvio il controller li richiamerà nella propria flash. I parametri dell'encoder salvati dopo un riavvio di un controller al quale non sia stato collegato un attuatore risulteranno quindi invalidi.

Trasferimento di un file di parametri al controller

Trasferendo un file di impostazione dei parametri ad un controller, i parametri dell'encoder verranno memorizzati in una memoria EEPROM collegata allo stesso encoder(escludendo i parametri nascosti, con accesso riservato ai tecnici IAI). Se il file di parametri è stato salvato da un controller scollegato dagli attuatori, verranno salvati sulla EEPROM dell'encoder dei parametri invalidi.

Avvertenze sulle posizioni aggiuntive

I controller più recenti possono memorizzare fino a 20000 posizioni. Dotando uno di questi controller della batteria di backup per la memorizzazione dei dati, le posizioni dalla 10001 alla 20000 verranno comunque salvate sulla flash ROM. Conseguentemente le modifiche non salvate per le posizioni dalla 10001 alla 20000 andranno perse nel caso di spegnimento del controller. Verranno invece mantenute grazie alla batteria di backup le posizioni dalla 1 alla 10000. Alla riaccensione del controller verranno caricate le posizioni memorizzate in precedenza nella flash ROM, sovrascrivendo eventuali modifiche.

(Nota) Salvare sempre i parametri del controller con gli attuatori cablati correttamente.

Avvertenze sui parametri aggiuntivi

Nei controller dotati di Gateway RC è stato inoltre realizzato un aumento dei parametri a disposizione.

	Numero di parametri	
	Senza RC Gateway	Con RC Gateway
I/O	400	600
Comune Assi	300	400
Specifico Asse	220	250
Driver	97	97
Encoder	30	30
Componenti I/O	82	82
Altri	120	200

Salvando i parametri di un controller dotato di gateway RC si otterrà quindi un file di dimensioni maggiori rispetto a quello ottenuto su controller precedenti privi di questa funzionalità. Il trasferimento di un set di parametri tra controller di generazioni differenti è limitato ai soli parametri comuni ad entrambe le versioni, come rappresentato nella figura 2.6.

Parametri supportati dai controller privi delle espansioni di memoria
Trasferiti
CASO [A]
Parametri supportati dai controller privi delle espansioni di memoria
Questi parametri non saranno modificati. Correggerli manualmente se necessario.
Parametri non supportati
Parametri supportati dai controller privi delle espansioni di memoria
Trasferiti
CASO [B]
Parametri supportati dai controller privi delle espansioni di memoria
Parametri non supportati
Non trasferiti nel controller

Figura 2.6: Schematizzazione del trasferimento parametri nelle due varianti.

2.2 I dati nel linguaggio SEL

2.2.1 Valori e Simboli

Il linguaggio SEL si basa sia su variabili che su simboli.

Lista dei valori e dei simboli usabili

Funzione	Range Globale	Range Locale	Note
Ingressi	0 ≈ 299		
Uscite	300 ≈ 599		
Flag	600 ≈ 899	900 ≈ 999	
Variabili Intere	200 ≈ 299	1 ≈ 99	IN, etc usano la 99
	1200 ≈ 1299	1001 ≈ 1099	
Variabili Reali	300 ≈ 399	100 ≈ 199	PPUT, etc usano la 199
	1300 ≈ 1399	1100 ≈ 1199	
Stringhe	300 ≈ 999	1 ≈ 299	
Tag		0 ≈ 256	
Subroutine		1 ≈ 99	
Coord. lavoro	0 ≈ 31		Solo SCARA
Coord. utensile	0 ≈ 127		Solo SCARA
Interferenza	1 ≈ 10		Solo SCARA
Zone	1 ≈ 4		Solo Assi
Pallet		1 ≈ 10	
Numero Asse	1 ≈ 6		
Codifica Asse	0 ≈ 111111		
Posizioni (Gw RC)	1 ≈ 20000		
Posizioni (No Gw RC)	1 ≈ 4000		
Programmi (Gw RC)	1 ≈ 128		
Programmi (No Gw RC)	0 ≈ 64		
Istruzioni (Gw RC)	1 ≈ 9999		
Istruzioni (No Gw RC)	1 ≈ 6000		
Task Level	NORM/CRIT	1 ≈ 299	
Porta Seriale	1 ≈ 2		
Timer di attesa		1	
Timer concorrenti		16	
Timer Ladder		Flag locali	
Ingressi Virtuali	7000 ≈ 7299		
Uscite Virtuali	7300 ≈ 7599		
Simboli	1000		

Caution

- Le variabili 99 e 199 sono soggette ad un uso speciale abbinato al alcuni comandi specifici del linguaggio SEL. Si eviti di utilizzarle per scopi generici.
- I valori riportati in tabella sono relativi alla massima capacità di gestione software delle variabili. Considerando elementi che hanno anche un limite fisico, quali ingressi ed uscite, si considera l'intersezione tra i due range.
- Un controller dotato di batteria di backup manterrà in memoria, anche se privo di alimentazione, i valori Globali delle variabili e dei flag.
- Le variabili ed i flag locali verranno azzerati al riavvio del programma.
- Range dei valori utilizzabili nel linguaggio SEL
  I numeri interi e reali sono soggetti a limitazioni come segue:

  1. Valori Numerici
     I controller SEL possono gestire valori numerici fino a otto cifre comprensivi di segno e punto decimale. Per gli interi il range è quindi da -9,999,999 a 99,999,999. Per i numeri reali non è previsto un limite di valore ma solo di cifre: il punto conta come cifra (Esempio: -0.12345) limitando se presente il numero delle cifre valide a sette.
  2. Valori delle posizioni
     Il limite di inserimento per i valori delle posizioni consiste in quattro cifre per la parte intera ed in tre cifre per la parte decimale, ovvero da -9999.999 a 9999.999 (il valore effettivo di limite fisico dipende dal modello di attuatore).

La maggior parte dei comandi del linguaggio SEL utilizza, nella propria sintassi, riferimenti a punti, variabili, ingressi, uscite, memorie interne o etichette; ognuno di questi elementi viene referenziato in modo univoco mediante un numero: i numeri disponibili per ogni famiglia di elementi sono stati presentati nella tabella della pagina precedente. Viene di seguito fornita una breve descrizione di tali elementi.

[**ingresso**]
: Rappresenta, dal punto di vista logico, lo stato del dispositivo (fotocellula, finecorsa, pulsante, etc.) collegato al corrispondente pin della scheda di I/O; può quindi avere valore 1 (stato logico ON) o 0 (stato logico OFF);

[**uscita**]
: Determina, secondo lo stato logico a cui viene impostata (ON o OFF), l'attivazione o disattivazione del dispositivo a cui è collegato il corrispondente pin della scheda di I/O;

[**flag (detto anche merker)**]
: Memoria ausiliaria della dimensione di un bit che può assumere valore logico 1 o 0;

[**variabile intera**]
: Memoria del controller che può assumere valori interi;

[**variabile reale**]
Memoria del controller che può assumere valori decimali; l'immissione diretta di valori è l'imitata dalla disponibilità di un massimo di 8 cifre

[**etichetta**]
Riferimento ad una particolare linea di un programma, utilizzata per alterare il flusso del programma stesso;

[**subroutine**] Riferimento alla linea iniziale di un gruppo di istruzioni che può essere richiamato da più punti all'interno del programma stesso;

[**posizione**]
Insieme delle coordinate (in mm) che caratterizzano un punto verso il quale è possibile indirizzare il moto degli assi gestiti dal controller;

[**programma**]
Insieme di istruzioni successive che gestiscono il flusso delle operazioni che il controller deve compiere;

[**task**]
Programma in esecuzione; più programmi possono venire eseguiti in parallelo, secondo il numero massimo dipendente dal modello del controller; i task possono scambiarsi dati mediante l'utilizzo di flag e variabili globali;

[**timer**]
Strumento di programmazione utilizzato per impostare pause nell'esecuzione di un task.

I/O Digitali

Distinti in ingressi ed in uscite. Nel caso standard il range attribuito va dalla porta No. 0 alla porta No. 31 per gli ingressi e dalla porta No. 300 alla porta No. 315 per le uscite.

I/O Virtuali

Il controller prevede un set di ingressi ed un set di uscite virtuali. Si consiglia la consultazione delle tabelle seguenti prima di un eventuale utilizzo.

Tabella 2.1: Ingressi Virtuali

No. Porta	Funzione
7000	Sempre OFF
7001	Sempre ON
7002	Allarme bassa tensione batteria di backup
7003	Tensione anormale batteria di sistema
7004	Uso proibito
7005	Uso proibito
7006 ≈ 7008	Errore di sistema
7009	Uso proibito
7010	Presenza evento critico per taglio alimentazione drive di potenza
7011	Richiesta pausa operazioni
7012	Presenza evento critico per la richiesta di pausa delle operazioni
7013	Presenza evento critico pausa e vincolo assi
7014	Uso proibito
7015	Bassa tensione batteria encoder abs asse 1
7016	Tensione anormale batteria encoder abs asse 1
7017	Bassa tensione batteria encoder abs asse 2
7018	Tensione anormale batteria encoder abs asse 2
7019	Bassa tensione batteria encoder abs asse 3
7020	Tensione anormale batteria encoder abs asse 3
7021	Bassa tensione batteria encoder abs asse 4
7022	Tensione anormale batteria encoder abs asse 4
7023	Bassa tensione batteria encoder abs asse 5
7024	Tensione anormale batteria encoder abs asse 5
7025	Bassa tensione batteria encoder abs asse 6
7026	Tensione anormale batteria encoder abs asse 6
7027 ≈ 7040	Uso proibito
7041 ≈ 7070	Uso proibito
7071	Controller in modalità AUTO
7072	Operazione automatica in corso
7073 ≈ 7100	Uso proibito
7101 ≈ 7228	Programma No. 1 ≈ 128 in esecuzione
7229 ≈ 7299	Uso proibito

Tabella 2.2: Uscite Virtuali

No. Porta	Funzione
7300	Presenza evento cancellazione operazioni
7301 ≈ 7380	Uso proibito
7381 ≈ 7399	Uso proibito
7400 ≈ 7599	Uso proibito

Flags

Diversamente dal suo significato comune, il termine *Flag* in programmazione acquisisce il valore di *Memoria*. I flag vengono comunemente utilizzati per impostare o resettare dei dati. Operano come relay ausiliari del controller.

I flag sono suddivisi in globali, accessibili da tutti i programmi, e locali, specifici di ogni singolo programma in esecuzione. Il valore dei flag locali viene perso ad ogni riavvio del controller, mentre quelli dei flag globali vengono mantenuti in memoria, preservati dalla batteria di backup.

Variabili

Variabile è un termine tecnico utilizzato in programmazione. Semplificando il concetto, una variabile è una scatola in cui si conserva un valore. Gli usi possibili delle variabili sono molteplici, e spaziano dalla memorizzazione di un valore all'esecuzione di operazioni matematiche.

Applicando il comando ADD 5 (somma) alla variabile 1 che già contiene il valore 2, 5 verrà sommato a 2 ottenendo 7, memorizzato nella variabile 1.

Tipologie di variabili Le variabili utilizzabili nel linguaggio SEL sono classificate in due differenti tipologie:

1. Variabili Intere
   Queste variabili non possono gestire elementi decimali.
2. Variabili Reali
   Queste variabili possono gestire elementi decimali.

Utilizzo dell'asterisco (*) Numerosi comandi del linguaggio SEL possono gestire, come operando, sia un valore numerico che il valore di una variabile; poiché le variabili sono a loro volta identificate da numeri, viene utilizzato l'operatore asterisco (*) per distinguerle quando possa nascere ambiguità.

Ad esempio, il comando di assegnamento (LET) prevede la variabile di destinazione come primo operando (utilizzata senza asterisco) e un secondo operando che può essere un numero o una variabile:

LET	205	217	Assegna alla variabile 205 il numero 217
LET	205	*217	Assegna alla variabile 205 il valore della variabile 217

L'operatore (*) è utilizzato anche per distinguere una variabile dal numero di una posizione:

MOVP	15	Muove gli assi al punto numero 15
MOVP	*15	Muove gli assi al punto corrispondente al valore della variabile 15

Tags

Un programma SEL è costituito da una serie di istruzioni che vengono eseguite sequenzialmente dal controller; l'ordine di esecuzione può essere alterato utilizzando istruzioni di salto e subroutine. Il termine *Tag* identifica un'etichetta, utilizzata allo stesso modo delle linguette attaccate alle pagine di un libro per facilitarne la consultazione. L'istruzione di salto (GOTO) forza il controllo a proseguire nell'esecuzione del programma dalla linea contraddistinta dall'etichetta specificata; il comando TAG consente appunto di associare un'etichetta alla linea in cui viene inserito.

```
XXXX   XXX   XXX
XXXX   XXX   XXX
XXXX   XXX   XXX
GOTO    28
YYYY   YYY   YYY
YYYY   YYY   YYY
TAG     28
```

Nell'esempio riportato, dopo l'esecuzione delle prime tre linee, l'istruzione di salto forza il controllo a passare alla linea contraddistinta dall'etichetta 28: le due righe YYYY non vengono eseguite. Evidentemente non è possibile specificare in uno stesso programma più di una linea con la stessa etichetta, mentre si possono introdurre istruzioni di salto ad una stessa etichetta da più linee.

Subroutines

L'altro metodo per alterare il normale svolgimento di un programma consiste nell'utilizzo di subroutine (sottocicli): una subroutine è un insieme di linee consecutive di codice comprese tra un'istruzione di inizio subroutine (BGSR) e una di fine (EDSR), e contraddistinta da un numero, specificato nella linea di inizio della subroutine stessa. Le subroutine sono esclusivamente locali, quindi valide solo all'interno del programma in cui sono state definite.

E' possibile richiamare una subroutine da qualsiasi punto del programma mediante l'istruzione EXSR, specificandone il numero. Il controllo passa ad eseguire la subroutine specificata, ma tornerà alla riga successiva a quella della chiamata una volta raggiunto il comando EDSR.

```
XXXX  XXX  XXX
EXSR   28
YYYY  YYY  YYY
YYYY  YYY  YYY
-----------------
BGSR   28
ZZZZ  ZZZ  ZZZ
ZZZZ  ZZZ  ZZZ
EDSR   28
```

Nell'esempio riportato, dopo la prima linea di codice, viene eseguita la subroutine 28, ovvero le due linee ZZZZ; raggiunta l'istruzione EDSR, il controllo passa alla prima istruzione YYYY. Come evidenziato, è possibile inserire, per aumentare la leggibilità, delle righe vuote prima delle istruzioni BGSR e dopo la EDSR, separando così anche visivamente le subroutine dal resto del codice. Come per le etichette, non è possibile specificare in uno stesso programma più di una subroutine con lo stesso numero, mentre è possibile richiamare la stessa subroutine da linee diverse. E' consigliabile, per una maggiore leggibilità, porre il ciclo principale in testa al programma, seguito dalle subroutine ordinate per numero crescente. L'ultima istruzione del ciclo principale deve essere EXIT (termine programma) o un salto (GOTO) ad una linea interna al ciclo stesso.

Simboli

Il linguaggio SEL consente di gestire le variabili ed i flag come Simboli. Per le metodologie di editazione dei simboli si faccia riferimento al manuale della Teaching Pendant(EN) e del software per una trattazione approfondita.

1. Simboli supportati
   I seguenti elementi possono essere definiti da un simbolo: Numero di una variabile, Numero di un flag, Numero di un tag, Numero di una subroutine, Numero di un programma, Numero di una posizione, Numero di un ingresso, Numero di un'uscita, Numero di un asse, costante.

2. Regole di definizione di un simbolo

   (a) La stringa alfanumerica che definisce un simbolo non può eccedere gli otto caratteri;
   - L'uso del trattino basso come primo carattere è consentito dalla versione 1.1.0.5 del software e dalla versione 1.04 della Teaching Pendant.
   - L'uso di caratteri ASCII compresi tra 21h e 7Eh come secondo carattere o successivo è consentito dalla versione di software 1.1.0.5 e dalla versione 1.04 della Teaching Pendant.
   - Lo stesso carattere ASCII ha diverse formulazioni nel Software e nella Teaching Pendant.

   (b) Non possono essere definiti simboli globali con lo stesso nome. Possono invece essere definiti due simboli locali con lo stesso nome purchè appartenenti a programmi diversi.

3. Numero massimo di simboli definibili: 1000;

4. Numero massimo di chiamate ai simboli nei programmi SEL: 5000, comprensivo di stringhe

Stringhe

Le stringhe alfanumeriche possono essere gestite tramite appositi comandi. Si rimanda alla sezione dei comandi presente in questo manuale.

Designazione degli assi

Il controller SEL può gestire (secondo il modello), fino un massimo di 6 assi. Alcuni comandi hanno effetto su uno solo di essi (comandi BASE, PPUT e PGET): la sintassi di tali comandi prevede un operando, che può assumere valori compresi tra 1 e 6, per specificare a quale asse debba essere applicato il comando stesso. I comandi che possono avere effetto su più di un asse per volta utilizzano invece, come operando, il cosiddetto *axis pattern*: esso consiste in una sequenza di 1 e 0 atta a specificare a quali assi deve essere riferito il comando stesso. Poiché il controllo può gestire fino a 6 assi, al massimo 6 bit possono essere utilizzati, ordinati per numero d'asse decrescente. La presenza di uno 0 nel pattern indica che il comando non deve essere applicato al corrispondente asse, mentre 1 indica che il comando deve essere applicato all'asse; gli 0 in testa al pattern possono non venire specificati.

Asse	6	5	4	3	2	1
Applicare	1	1	1	1	1	1
Non Applicare	0	0	0	0	0	0

Ad esempio il comando HOME 1001 (equivalente a HOME 001001) esegue la ricerca dell'origine per gli assi 1 e 4, mentre il comando HOME 100 esegue la ricerca origine per il solo asse 3.

Il pattern assi viene utilizzato dai comandi OFST, GRP, SVON, SVOF, HOME, JFWN, JFWF, JBWN, JBWF, STOP, PTST, PRED, CHVL, PBND, WZNA, WZNO, WZFA, WZFO.

2.2.2 Tabella Posizioni

Il linguaggio SEL si basa sui dati contenuti nella tabella posizioni e sui comandi contenuti nei programmi. Nella tabella delle posizioni vengono memorizzati i valori di posizione, le coordinate, velocità accelerazione e decelerazione.

No. Pos.	Asse 1[mm]	Asse 2[mm]	Asse 3[mm]	Asse 4[mm]	Vel[mm/s]	Acc[G]	Dcl[G]
1 ≈ 4000							

I valori di velocità, accelerazione e decelerazione impostabili variano in base all'attuatore installato. La definizione di questi valori nella tabella delle posizioni ottiene una priorità superiore rispetto ad altre definizioni impostate all'interno del programma.

Volendo gestire il profilo di moto differentemente in base ad eventi esterni, gestendolo interamente da programma, lasciare liberi questi campi nella tabella posizioni.

Lavorando con un attuatore rotante i valori sono processati in gradi, e non in millimetri. Se il parametro No. 1 degli Axis Specific Parameter, che definisce la tipologia dell'asse tra lineare e rotante, è impostato a 1, tutti i valori sono relativi ad un asse rotativo e quindi espressi in gradi.

2.2.3 Comandi

La caratteristica primaria del linguaggio SEL è la sua semplicità. Questo consente di evitare di introdurre un compilatore, e di gestire le operazioni ad alta velocità.

Struttura del linguaggio SEL

La tabella seguente illustra la struttura di un singolo passo.

Condizione(AND,OR)	Input	Comando	Operando 1	Operando 2	Output

1. Se la condizione di Input è soddisfatta, il comando impostato nella riga verrà eseguito. Se è presente un'impostazione di Output, al completamento dell'esecuzione del comando l'uscita corrispondente verrà impostata al valore ON. Se al contrario la condizione di Input non dovesse essere soddisfatta, il comando non verrà eseguito. Conseguentemente non accadrà nulla alla porta di uscita.
2. Se non è stato impostato nessun input, il comando verrà eseguito incondizionatamente.
3. Per usare la condizione con logica negata introdurre la N (Not) alla condizione.
4. Per l'input possono essere utilizzati ingressi, uscite e flag.
5. Sia l'operando 1 che l'operando 2 che l'output possono essere specificati indirettamente, tramite un simbolo.

Estensioni condizionali

Le condizioni di esecuzione di un comando possono essere combinate anche in strutture molto complesse.

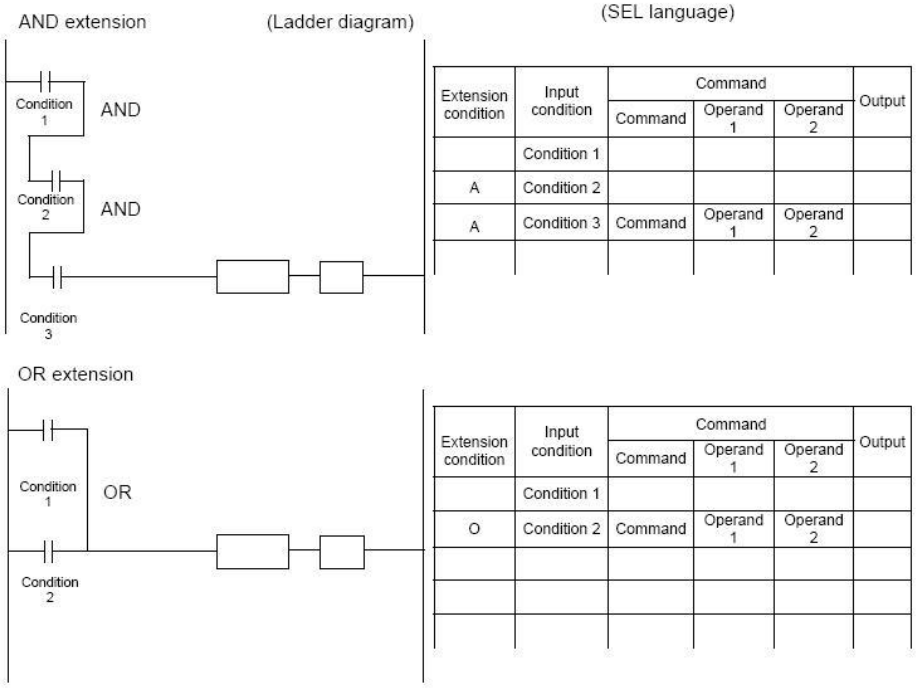


Figura 2.7: Costrutto AND e costrutto OR.

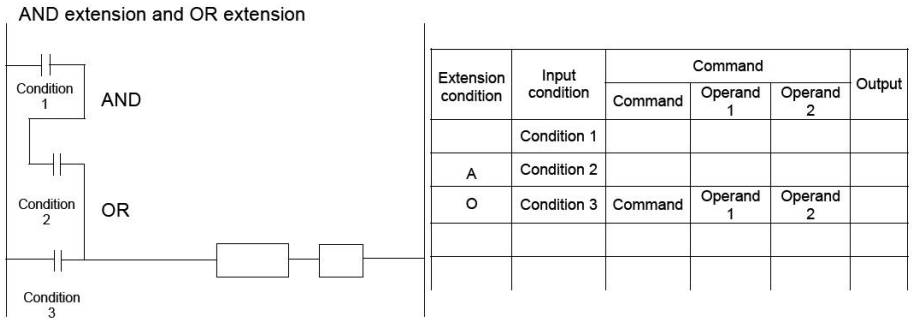


Figura 2.8: Costrutto AND - OR.

Capitolo 3

Spiegazione dei comandi

Indice

3.1 Comandi 38
3.1.1 Assegnamento di variabili 38
3.1.2 Operazioni Aritmetiche 40
3.1.3 Operazioni Funzionali 43
3.1.4 Operazioni Logiche 46
3.1.5 Comparazioni 49
3.1.6 Timer 50
3.1.7 Operazioni con I/O e Flag 52
3.1.8 Gestione dei programmi 61
3.1.9 Gestione delle operazioni 64
3.1.10 Gestione delle posizioni 67
3.1.11 Controllo degli attuatori 76
3.1.12 Movimentazione degli attuatori 101
3.1.13 IF strutturato 107
3.1.14 DO strutturato 108
3.1.15 Multi-Branch 109
3.1.16 Acquisizione informazioni sistema 110
3.1.17 Zone 111
3.1.18 Comunicazione 112
3.1.19 Operazioni sulle stringhe 113
3.1.20 Gestione pallet 114
3.1.21 Impostazione pallet 115
3.1.22 Movimenti su pallet 116
3.1.23 Pseudo-Ladder 117
3.1.24 Comandi estesi 118

3.1 Comandi

3.1.1 Assegnamento di variabili

- **LET (Assegna)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	LET	Numero Variabile	Dato	ZR

Il comando LET assegna il valore specificato nell'operando 2 alla variabile impostata nel campo operando 1. L'output verrà impostato ad ON assegnando 0 alla variabile definita nell'operando 1.

Esempio 1	LET	1	10	Assegna alla variabile 1 il valore 10
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2
	LET	3	10	Assegna alla variabile 3 il valore 10
	LET	*1	*3	Assegna il contenuto della variabile 3 (10) al contenuto della variabile 1 (variabile 2)

- **TRAN (Copia)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	TRAN	Numero Variabile	Numero Variabile	ZR

Il comando TRAN assegna il contenuto della variabile contenuta nell'operando 2 valore specificato nel campo operando 2 alla variabile impostata nel campo operando 1. L'output verrà impostato ad ON assegnando 0 alla variabile contenuta nell'operando 1.

Esempio 1	TRAN	1	2	Assegna alla variabile 1 il contenuto della variabile 2
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2
	LET	2	3	Assegna alla variabile 2 il valore 3
	LET	3	4	Assegna alla variabile 3 il valore 4
	LET	4	10	Assegna alla variabile 4 il valore 10
	TRAN	*1	*3	Assegna il contenuto della variabile 3 (10) alla variabile 2 (*1)

Le variabili cambiano come segue:

- **CLR (Cancella)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	CLR	Numero Variabile	Numero Variabile	ZR

Il comando CLR cancella le variabili a partire da quella definita nell'operando 1 fino ad arrivare a quella contenuta nell'operando 2. Il contenuto delle variabili cancellate verrà posto al valore 0. L'output verrà impostata ad ON all'azzeramento della variabile definita nell'operando 1.

Esempio 1	CLR	1	5	Cancella le variabili dalla 1 alla 5
Esempio 2	LET	1	10	Assegna alla variabile 1 il valore 10
	LET	2	20	Assegna alla variabile 2 il valore 20
	CLR	*1	*2	Cancella le variabili dalla 10 (*1) alla 20 (*2)

3.1.2 Operazioni Aritmetiche

- **ADD (Somma)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	ADD	Numero Variabile	Dato	ZR

Il comando ADD somma il valore specificato nell'operando 2 al contenuto della variabile 1, e lo assegna alla variabile 1. L'output verrà impostato ad ON qualora il risultato dell'operazione sia pari a 0.

Esempio 1	LET	1	3	Assegna alla variabile 1 il valore 3
	ADD	1	2	Somma 2 al contenuto della variabile 1 (3). Assegna alla variabile 1 il valore 5
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2
	LET	2	3	Assegna alla variabile 2 il valore 3
	LET	3	10	Assegna alla variabile 3 il valore 10
	ADD	*1	*3	Somma il contenuto della variabile 3(2) al contenuto della variabile 1 (variabile 2). Assegna alla variabile 2 il valore 5

- **SUB (Sottrae)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SUB	Numero Variabile	Dato	ZR

Il comando SUB sottrae il valore specificato nell'operando 2 al contenuto della variabile 1, e assegna il risultato alla variabile 1. L'output verrà impostato ad ON qualora il risultato dell'operazione sia pari a 0.

Esempio 1	LET	1	3	Assegna alla variabile 1 il valore 3
	SUB	1	2	Sottrae 2 al contenuto della variabile 1 (3). Assegna alla variabile 1 il valore 1
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2
	LET	2	3	Assegna alla variabile 2 il valore 3
	LET	3	10	Assegna alla variabile 3 il valore 10
	SUB	*1	*3	Sottrae il contenuto della variabile 3(2) al contenuto della variabile 1 (variabile 2). Assegna alla variabile 2 il valore 1

- **MULT (Moltiplica)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MULT	Numero Variabile	Dato	ZR

Moltiplica il contenuto della variabile specificata nell'operando 1 per il valore specificato nell'operando 2. Assegna il risultato alla variabile specificata nell'operando 1. L'output verrà impostato ad ON qualora il risultato dell'operazione sia pari a 0.

Esempio 1	LET	1	3	Assegna alla variabile 1 il valore 3
	MULT	1	2	Moltiplica il valore contenuto nella variabile 1 per 2. Assegna alla variabile 1 il valore 6.
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2
	LET	2	3	Assegna alla variabile 2 il valore 3
	LET	3	10	Assegna alla variabile 3 il valore 10
	MULT	*1	*3	Moltiplica per il contenuto della variabile 3(2) il contenuto della variabile 1 (variabile 2). Assegna alla variabile 2 il valore 6.

- **DIV (Divide)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DIV	Numero Variabile	Dato	ZR

Divide il contenuto della variabile specificata nell'operando 1 per il valore specificato nell'operando 2. Assegna il risultato alla variabile specificata nell'operando 1. L'output verrà impostato ad ON qualora il risultato dell'operazione sia pari a 0. Si noti che lavorando con variabile intere il risultato verrà approssimato escludendo i decimali.

Esempio 1	LET	1	6	Assegna alla variabile 1 il valore 6.
	DIV	1	2	Divide per 2 il contenuto della variabile 1(6). Assegna alla variabile 1 il valore 3.
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2.
	LET	2	6	Assegna alla variabile 2 il valore 6.
	LET	3	2	Assegna alla variabile 3 il valore 2.
	DIV	*1	*3	Divide il contenuto della variabile 1(variabile 2) per il contenuto della variabile 3 (2). Assegna alla variabile 2 il valore 3.

- **MOD (Resto della divisione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MOD	Numero Variabile	Dato	ZR

Divide il contenuto della variabile specificata nell'operando 1 per il valore specificato nell'operando 2. Assegna il resto della divisione alla variabile specificata nell'operando 1. L'output verrà impostato ad ON qualora il risultato dell'operazione sia pari a 0. **Attenzione:** il comando MOD è applicabile solo a variabili intere.

Esempio 1	LET	1	7	Assegna alla variabile 1 il valore 7.
	MOD	1	3	Divide per 3 il contenuto della variabile 1(7). Assegna alla variabile 1 il valore 1 ($7 \div 3 = 2$ con il resto di 1).
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2.
	LET	2	7	Assegna alla variabile 2 il valore 7.
	LET	3	3	Assegna alla variabile 3 il valore 3.
	DIV	*1	*3	Divide il contenuto della variabile 1(variabile 2) per il contenuto della variabile 3 (3). Assegna alla variabile 2 il valore 1.

3.1.3 Operazioni Funzionali

- **SIN (Funzione seno)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SIN	Numero Variabile	Dato	ZR

Assegna il seno del dato contenuto nell'operando 2 alla variabile specificata nell'operando 1. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0. La variabile di destinazione impostata nell'operando 1 deve essere Reale. Il dato contenuto nell'operando 2 va espresso in radianti.

Esempio 1	SIN	100	0.523599	Assegna alla variabile 100 il seno di 0.523599 (0.5).
Esempio 2	LET	1	100	Assegna alla variabile 1 il valore 100.
	LET	101	30	Assegna alla variabile 101 il valore 30.
	MULT	101	3.141592	
	DIV	101	180	Conversione da gradi (30) in radianti.
	SIN	*1	*101	Assegna il seno del contenuto della variabile 101 al contenuto della variabile 1 (variabile 100).

- **COS (Funzione coseno)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	COS	Numero Variabile	Dato	ZR

Assegna il coseno del dato contenuto nell'operando 2 alla variabile specificata nell'operando 1. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0. La variabile di destinazione impostata nell'operando 1 deve essere Reale. Il dato contenuto nell'operando 2 va espresso in radianti.

Esempio 1	COS	100	1.047197	Assegna alla variabile 100 il coseno di 1.047197 (0.5).
Esempio 2	LET	1	100	Assegna alla variabile 1 il valore 100.
	LET	101	60	Assegna alla variabile 101 il valore 60.
	MULT	101	3.141592	
	DIV	101	180	Conversione da gradi (60) in radianti.
	COS	*1	*101	Assegna il coseno del contenuto della variabile 101 al contenuto della variabile 1 (variabile 100).

• **TAN (Funzione tangente)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	TAN	Numero Variabile	Dato	ZR

Assegna la tangente del dato contenuto nell'operando 2 alla variabile specificata nell'operando 1. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0. La variabile di destinazione impostata nell'operando 1 deve essere Reale. Il dato contenuto nell'operando 2 va espresso in radianti.

Esempio 1	TAN	100	0.785398	Assegna alla variabile 100 la tangente di 1.047197 (0.5).
Esempio 2	LET	1	100	Assegna alla variabile 1 il valore 100.
	LET	101	45	Assegna alla variabile 101 il valore 45.
	MULT	101	3.141592	
	DIV	101	180	Conversione da gradi (45) in radianti.
	TAN	*1	*101	Assegna la tangente del contenuto della variabile 101 al contenuto della variabile 1 (variabile 100).

• **ATN (Funzione inverso della tangente)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	ATN	Numero Variabile	Dato	ZR

Assegna l'inverso della tangente del dato contenuto nell'operando 2 alla variabile specificata nell'operando 1. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0. La variabile di destinazione impostata nell'operando 1 deve essere Reale. L'inverso della tangente sarà espresso in radianti.

Esempio 1	ATN	100	1	Assegna alla variabile 100 L'inverso della tangente di 1 (0.785398).
Esempio 2	LET	1	100	Assegna alla variabile 1 il valore 100.
	LET	101	1	Assegna alla variabile 101 il valore 60.
	ATN	*1	*101	Assegna l'inverso della tangente del contenuto della variabile 101 al contenuto della variabile 1 (variabile 100).

- **SQR (Funzione radice quadrata)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SQR	Numero Variabile	Dato	ZR

Assegna la radice quadrata del dato contenuto nell'operando 2 alla variabile specificata nell'operando 1. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0.

Esempio 1	SQR	1	4	Assegna alla variabile 1 la radice quadrata di 4.
Esempio 2	LET	1	10	Assegna alla variabile 1 il valore 10.
	LET	2	4	Assegna alla variabile 2 il valore 4.
	SQR	*1	*2	Assegna la radice quadrata del contenuto della variabile 2 (4) al contenuto della variabile 1 (variabile 10).

3.1.4 Operazioni Logiche

- **AND (AND Logico)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	AND	Numero Variabile	Dato	ZR

Assegna alla variabile specificata nell'operando 1 il risultato di un AND logico tra il suo contenuto ed il valore specificato nell'operando 2. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0.

Esempio 1	LET	1	204	Assegna alla variabile 1 il valore 204.
	AND	1	170	Assegna alla variabile 1 il risultato dell'operazione di AND logico (136) tra il contenuto della variabile 1 (204) ed il valore 170.
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2.
	LET	2	204	Assegna alla variabile 2 il valore 204.
	LET	3	170	Assegna alla variabile 3 il valore 170.
	AND	*1	*3	Assegna alla variabile 2 (contenuto della variabile 1) il risultato dell'operazione di AND logico (136) tra il contenuto della variabile 1 (204) ed il valore 170.

	Decimale	Binario
	204	11001100
AND	170	10101010
	136	10001000

- **OR (OR Logico)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	OR	Numero Variabile	Dato	ZR

Assegna alla variabile specificata nell'operando 1 il risultato di un OR logico tra il suo contenuto ed il valore specificato nell'operando 2. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0.

Esempio 1	LET	1	204	Assegna alla variabile 1 il valore 204.
	OR	1	170	Assegna alla variabile 1 il risultato dell'operazione di OR logico (238) tra il contenuto della variabile 1 (204) ed il valore 170.
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2.
	LET	2	204	Assegna alla variabile 2 il valore 204.
	LET	3	170	Assegna alla variabile 3 il valore 170.
	OR	*1	*3	Assegna alla variabile 2 (contenuto della variabile 1) il risultato dell'operazione di OR logico (238) tra il contenuto della variabile 1 (204) ed il valore 170.

	Decimale	Binario
	204	11001100
OR	170	10101010
	238	11101110

• **EOR (OR esclusivo Logico)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	EOR	Numero Variabile	Dato	ZR

Assegna alla variabile specificata nell'operando 1 il risultato di un OR esclusivo logico tra il suo contenuto ed il valore specificato nell'operando 2. La variabile Output verrà impostata ad ON qualora il risultato sia pari a 0.

Esempio 1	LET	1	204	Assegna alla variabile 1 il valore 204.
	EOR	1	170	Assegna alla variabile 1 il risultato dell'operazione di OR esclusivo logico (102) tra il contenuto della variabile 1 (204) ed il valore 170.
Esempio 2	LET	1	2	Assegna alla variabile 1 il valore 2.
	LET	2	204	Assegna alla variabile 2 il valore 204.
	LET	3	170	Assegna alla variabile 3 il valore 170.
	EOR	*1	*3	Assegna alla variabile 2 (contenuto della variabile 1) il risultato dell'operazione di OR esclusivo logico (102) tra il contenuto della variabile 1 (204) ed il valore 170.

	Decimale	Binario
	204	11001100
EOR	170	10101010
	238	01100110

3.1.5 Comparazioni

- **CPΞΞ (Comparazione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	CPΞΞ	Numero Variabile	Dato	Vedi Nota

La variabile di Output verrà impostata ad ON qualora il risultato della comparazione soddisfi la condizione scelta. I valori delle due variabili non cambieranno. L'output verrà impostato ad OFF nel caso di condizione non soddisfatta.
(Nota) Durante l'esecuzione del comando l'uscita sarà impostata ad OFF.

CPΞΞ

- ↘ EQ ... Operando 1 = Operando 2
- ↘ NE ... Operando 1 ≠ Operando 2
- ↘ GT ... Operando 1 > Operando 2
- ↘ GE ... Operando 1 ≥ Operando 2
- ↘ LT ... Operando 1 < Operando 2
- ↘ LE ... Operando 1 ≤ Operando 2

Esempio 1	LET	1	10		Assegna alla variabile 1 il valore 10.
	CPEQ	1	10	600	Imposta ad ON il flag 600 se il contenuto della variabile 1 è uguale a 10.
r600	ADD	2	1		Aggiunge 1 alla variabile 2 se il flag 600 è ON.
Esempio 2	LET	1	2		Assegna alla variabile 1 il valore 2.
	LET	2	10		Assegna alla variabile 2 il valore 204.
	LET	3	10		Assegna alla variabile 3 il valore 170.
	CPNE	*1	*3	310	Imposta ad ON l'uscita 310 se il contenuto della variabile 2 è diverso dal contenuto della variabile 3.

3.1.6 Timer

- **TIMW (Timer)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	TIMW	Tempo	Uso Proibito	TU

Arresta il programma e attende per il tempo specificato nell'operando 1. Il range di accettazione è da 0.01 a 99 e il valore deve essere espresso in secondi. L'output verrà impostato ad ON quando sarà trascorso il tempo impostato, e il programma proseguirà al passo successivo.

Esempio 1	TIMW	1.5		Attesa di 1.5 secondi.
Esempio 2	LET TIMW	1 *1	10	Assegna alla variabile 1 il valore 10. Attesa di 10 secondi (contenuto della variabile 1).

- **TIMC (Cancella i Timer)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	TIMC	Numero programma	Uso Proibito	CP

Cancella i timer relativi ad un programma in esecuzione in parallelo.
(Nota) I timer relativi alle istruzioni TIMW, WTON, WTOF e READ possono essere cancellati. In assenza di un tempo di attesa esplicito, il programma ne supporrà uno infinito.

Esempio 1	TIMC	10		Cancella i timer relativi al programma 10.
Esempio 2	LET TIMC	1 *1	10	Assegna alla variabile 1 il valore 10. Cancella i timer relativi al programma 10.

- **GTTM (Ottiene un Timer)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GTTM	Numero variabile	Uso Proibito	CP

Legge il tempo di sistema e lo trascrive nella variabile specificata nell'operando 1. Il tempo è espresso in decine di millisecondi.
(Nota) Non è previsto un riferimento di base per questo comando. Per valutare il tempo trascorso si deve prevedere una duplice chiamata dell'istruzione.

Esempio 1	GTTM	1		Assegna alla variabile 1 il tempo di sistema.
	ADD	1	500	Imposta un tempo finale 5 secondi dopo.
	GTTM	2		Assegna alla variabile 2 il tempo di sistema.
	DWLE	2	*1	Verifica che non siano ancora trascorsi 5 secondi.
	...			Istruzione da ripetere ciclicamente per 5 secondi.
	GTTM	2		Assegna alla variabile 2 il tempo di sistema aggiornato.
	EDDO			
Esempio 2	LET	1	10	Assegna alla variabile 1 il valore 10.
	GTTM	*1		Assegna alla variabile 10 (il tempo di sistema.

3.1.7 Operazioni con I/O e Flag

- **BTΞΞ (Impostazione uscite)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	BTΞΞ	Output, Flag	(Output, Flag)	CP

Assegna lo stato logico ON/OFF dall'uscita o dal flag specificato nell'operando 1 fino all'uscita o al flag dell'operando 2.
(Nota) Non è possibile modificare lo stato delle uscite di sistema con quest'istruzione.
BTΞΞ

↘ ON ... Imposta lo stato a ON
↘ OFF ... Imposta lo stato a OFF
↘ NT ... Inverte lo stato attuale

Esempio 1	BTON	300		Imposta a ON l'uscita 300.
Esempio 2	BTOF	300	307	Imposta a OFF le uscite dalla 300 alla 307.
Esempio 3	LET	1	600	Assegna 600 alla variabile 1.
	BTNT	*1		Inverte lo stato del contenuto della variabile 1.
Esempio 4	LET	1	600	Assegna alla variabile 1 il valore 600.
	LET	2	607	Assegna alla variabile 2 il valore 607.
	BTON	*1	*2	Imposta a ON i flag dal 600 al 607.

• **BTPN (Accensione temporizzata uscita)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	BTPN	Output, Flag	Timer	CP

Assegna lo stato logico ON/OFF dall'uscita o dal flag specificato nell'operando 1 per un intervallo di tempo specificato nell'operando 2. Impostata ad ON l'uscita il programma procederà all'esecuzione delle istruzioni successive.
(Nota 1) Applicando quest'istruzione su un'uscita già impostata ad ON, questa manterrà il suo stato e verrà successivamente impostata ad OFF trascorso il tempo di attesa impostato.
(Nota 2) Se il programma termina mentre l'uscita è ancora ad ON, questa verrà impostata immediatamente ad OFF.
(Nota 3) Quest'istruzione non verrà cancellata da un TIMC.
(Nota 4) Ogni programma può gestire al massimo 16 timer contemporaneamente.
(Nota 5) Non è possibile applicare questo comando alle uscite di sistema.
(Nota 6) Questo comando non può essere utilizzato per generare una costante di tempo.

Esempio 1	BTPN	300	1	Imposta a ON l'uscita 300 per 1 secondo.
Esempio 2	BTPN	600	10	Imposta a ON il flag 600 per 10 secondi.

• **BTPF (Spegnimento temporizzato uscita)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	BTPF	Output, Flag	Timer	CP

Assegna lo stato logico ON/OFF dall'uscita o dal flag specificato nell'operando 1 per un intervallo di tempo specificato nell'operando 2. Impostata ad ON l'uscita il programma procederà all'esecuzione delle istruzioni successive.
(Nota 1) Applicando quest'istruzione su un'uscita già impostata ad OFF, questa manterrà il suo stato e verrà successivamente impostata ad ON trascorso il tempo di attesa impostato.
(Nota 2) Se il programma termina mentre l'uscita è ancora ad OFF, questa verrà impostata immediatamente ad ON.
(Nota 3) Quest'istruzione non verrà cancellata da un TIMC.
(Nota 4) Ogni programma può gestire al massimo 16 timer contemporaneamente.
(Nota 5) Non è possibile applicare questo comando alle uscite di sistema.
(Nota 6) Questo comando non può essere utilizzato per generare una costante di tempo.

Esempio 1	BTPF	300	1	Imposta a OFF l'uscita 300 per 1 secondo.
Esempio 2	BTPF	600	10	Imposta a OFF il flag 600 per 10 secondi.

• **WT☰ (Attesa I/O o flag)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	WT	I/O, Flag	(Time)	TU

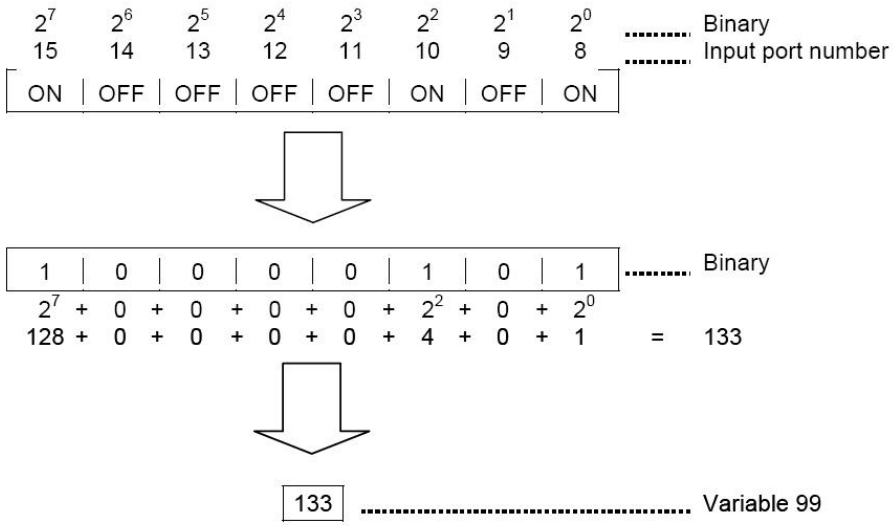
Attende un fronte allo stato logico ON/OFF dell'ingresso uscita o flag specificato nell'operando 1 per un intervallo massimo di tempo specificato nell'operando 2. L'Output verrà impostata ad ON se il tempo di attesa dovesse scadere prima del fronte desiderato.
(Nota 1) Non è possibile operare sui flag locali con quest'istruzione.
WT☰
↘ ON ... Attende che l'ingresso uscita o flag desiderato sia impostato ad ON.
↘ OFF ... Attende che l'ingresso uscita o flag desiderato sia impostato ad OFF.

Esempio 1	WTON	15		Attende un fronte di salita sull'ingresso 15.
Esempio 2	WTOF	308	10	Attende un fronte di discesa sull'uscita 308 per 10 secondi.

- **IN (Lettura I/O o flag da binario)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	IN	I/O, Flag	I/O, Flag	CC

Interpreta il codice binario letto dall'ingresso uscita o flag specificato nell'operando 1 all'ingresso uscita o flag specificato nell'operando 2. Salva il risultato nella variabile 99.



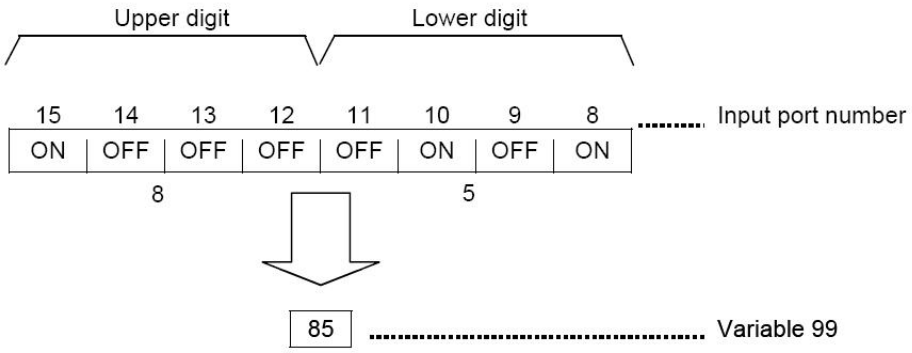
(Nota 1) Possono essere letti al massimo 32 input alla volta.
(Nota 2) Se leggendo 32 input il bit più significativo fosse impostato ad ON, il valore salvato nella variabile 99 sarebbe negativo.
(Nota 3) E' possibile modificare il formato utilizzato in lettura attraverso il comando FMIO.

Esempio 1	IN	8	15	Salva nella variabile 99 la conversione in decimale della lettura binaria dall'ingresso 8 al 15.
Esempio 2	LET	1	8	Assegna 8 alla variabile 1.
	LET	2	15	Assegna 15 alla variabile 2.
	IN	*1	*2	Salva nella variabile 99 la conversione in decimale della lettura binaria dall'ingresso 8 al 15.

- **INB (Lettura I/O o flag come un BCD)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	INB	I/O, Flag	Numero cifre	CC

Interpreta il codice BCD letto dall'ingresso uscita o flag specificato nell'operando 1 per un numero di cifre pari al valore specificato nell'operando 2. Salva il risultato nella variabile 99.



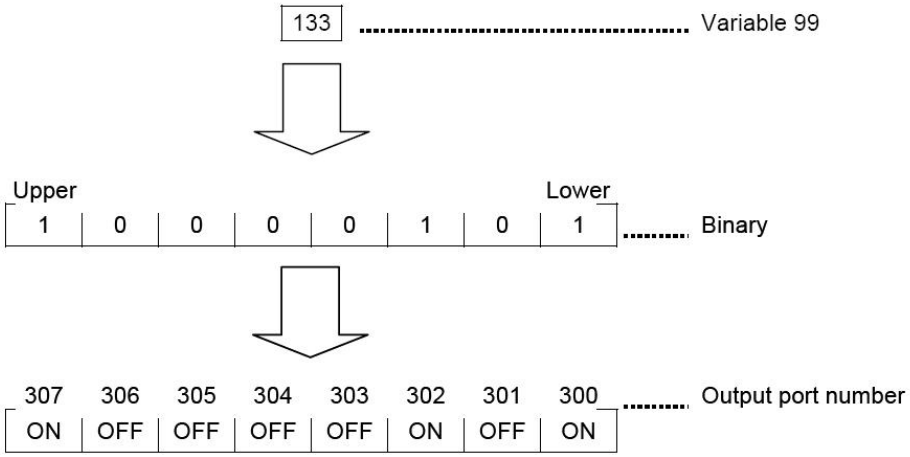
(Nota 1) Possono essere lette al massimo 8 cifre (32 input) alla volta.
(Nota 2) Verranno utilizzati tanti input output o flag quante sono le cifre da leggere, moltiplicate per 4.
(Nota 3) E' possibile modificare il formato utilizzato in lettura attraverso il comando FMIO.

Esempio 1	INB	8	2	Salva nella variabile 99 la conversione in decimale della lettura BCD dall'ingresso 8 al 15.
Esempio 2	LET	1	8	Assegna 8 alla variabile 1.
	LET	2	2	Assegna 2 alla variabile 2.
	INB	*1	*2	Salva nella variabile 99 la conversione in decimale della lettura BCD dall'ingresso 8 al 15.

- **OUT (Scrittura Output o flag in binario)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	OUT	Output, Flag	Output, Flag	CC

Interpreta il valore espresso nella variabile 99 e lo converte in codice binario, impostando di conseguenza lo stato delle uscite o dei flag a partire dal valore specificato nell'operando 1, fino all'uscita o flag specificato nell'operando 2.



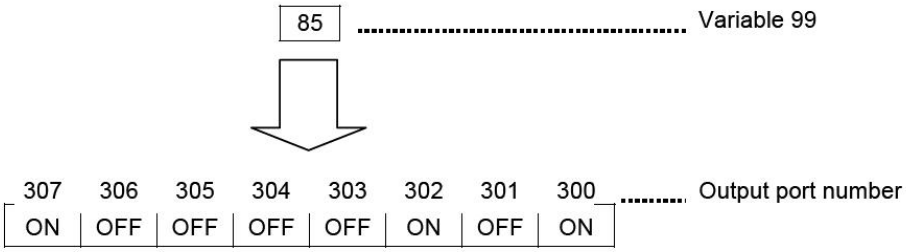
(Nota 1) Possono essere scritti al massimo 32 output alla volta.
(Nota 2) E' possibile modificare il formato utilizzato in lettura attraverso il comando FMIO.

Esempio 1	OUT	300	307	Legge dalla variabile 99 e converte il valore in binario. Imposta di conseguenza le uscite dalla 300 alla 307.
Esempio 2	LET	1	300	Assegna 300 alla variabile 1.
	LET	2	307	Assegna 307 alla variabile 2.
	OUT	*1	*2	Legge dalla variabile 99 e converte il valore in binario. Imposta di conseguenza le uscite dalla 300 alla 307.

- **OUTB (Scrittura Output o flag come un BCD)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	OUTB	I/O, Flag	Numero cifre	CC

Interpreta il valore salvato nella variabile 99 e lo converte in codice BCD, impostando di conseguenza le uscite o flag a partire da quella specificata nell'operando 1 per un numero di cifre pari al valore specificato nell'operando 2.



(Nota 1) Possono essere scritte al massimo 8 cifre (32 output) alla volta.
(Nota 2) Verranno utilizzati tanti output o flag quante sono le cifre da leggere, moltiplicate per 4.
(Nota 3) E' possibile modificare il formato utilizzato in lettura attraverso il comando FMIO.

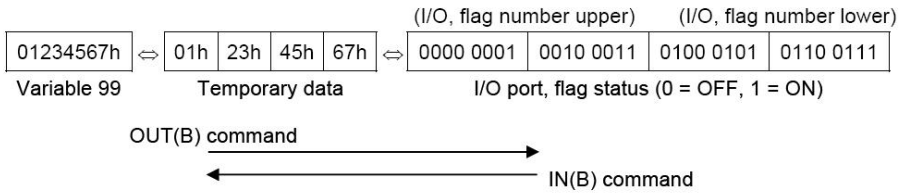
Esempio 1	OUTB	300	2	Legge dalla variabile 99 un valore decimale e lo converte in BCD dall'output 300 al 307.
Esempio 2	LET	1	300	Assegna 300 alla variabile 1.
	LET	2	2	Assegna 2 alla variabile 2.
	OUTB	*1	*2	Legge dalla variabile 99 un valore decimale e lo converte in BCD dall'output 300 al 307.

• **FMIO (Imposta il formato dei comandi IN, INB, OUT, OUTB)**

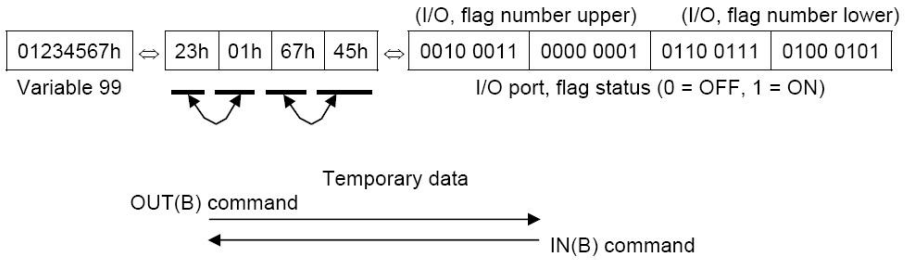
Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	FMIO	Formato	Uso Proibito	CP

Imposta il formato dei dati in lettura o scrittura delle porte I/O e dei flag abbinato ai comandi IN, INB, OUT, OUTB.

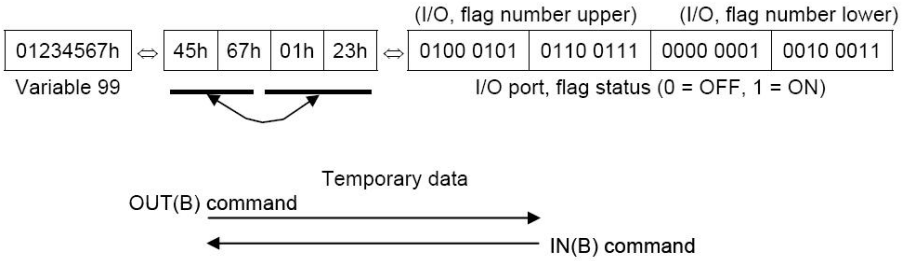
Operando 1 = 0 (Default)
I dati vengono letti o scritti come sono, senza inversioni.



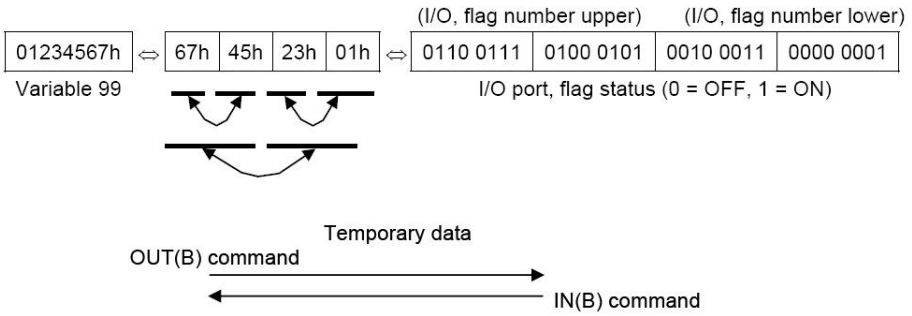
Operando 1 = 1
I dati vengono letti o scritti scambiando ogni 16 bit gli 8 bit meno significativi con gli 8 bit più significativi.



Operando 1 = 2
I dati vengono letti o scritti scambiando ogni 32 bit i 16 bit meno significativi con i 16 bit più significativi.



Operando 1 = 3
I dati vengono letti o scritti scambiando ogni 32 bit i 16 bit meno significativi con i 16 bit più significativi, ed ogni 16 bit gli 8 bit meno significativi con gli 8 bit più significativi.



3.1.8 Gestione dei programmi

• **GOTO (Jump)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GOTO	Numero Tag	Uso Proibito	CP

Rimanda alla posizione indicata dal Tag scelto.
(Nota 1) Un GOTO è valido solo all'interno del programma.
(Nota 2) Non creare un programma contenente un loop indefinito di movimenti usando la sintassi GOTO-TAG; produrrebbe un accumulo di errori di conversione delle coordinate.

Esempio 1	TAG	1	Imposta il Tag numero 1.
	...		
	GOTO	1	Salta al Tag 1.

L'uso del comando GOTO per uscire da una delle sintassi elencate in seguito è proibito:

1. IFΞΞ oppure ISΞΞ e EDIF
2. DWΞΞ e EDDO
3. SLCT e EDSL
4. BGSR e EDSR
5. BGPA e EDPA

• **TAG (Dichiarazione Etichetta)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	TAG	Numero Tag	Uso Proibito	CP

Assegna alla riga un'etichetta avente numero pari al valore inserito nell'operando 1.

• **EXSR (Esecuzione subroutine)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	EXSR	Numero Subroutine	Uso Proibito	CP

Esegue la subroutine impostata nell'operando 1. Al massimo possono venire eseguite 15 subroutine annidate.
(Nota) Questo comando è valido solo per subroutine presenti all'interno del programma.

Esempio 1	EXSR	1		Esegue la subroutine numero 1.
	...			
	...			
	EXIT			
	BGSR	1		Crea subroutine numero 1.
	...			
	...			
	EDSR			Termina la subroutine numero 1.
Esempio 2	LET	1	10	Assegna 10 alla variabile 1.
	EXSR	*1		Avvia la subroutine numero 10 (*1).

• **BGSR (Dichiarazione subroutine)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	BGSR	Numero Subroutine	Uso Proibito	CP

Dichiara la subroutine impostata nell'operando 1.
(Nota) E' proibito l'uso di un GOTO per uscire dalla subroutine.

• **EDSR (Termine subroutine)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	EDSR	Uso Proibito	Uso Proibito	CP

Dichiara il termine di una subroutine. Questo comando è sempre richiesto alla fine di una subroutine. Dopo questa istruzione il programma proseguirà dalla riga successiva a quella contenente l'istruzione EXSR.

3.1.9 Gestione delle operazioni

• **EXIT (Termine programma)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	EXIT	Uso Proibito	Uso Proibito	CP

Termina il programma. Se l'ultimo step è stato raggiunto e non contiene un'istruzione EXIT il programma ricomincerà dalla prima riga.
(Nota) Stato delle variabili al termine del programma:

- Output ⇒ Mantenuti
- Input ⇒ Azzerati
- Variabili Locali ⇒ Azzerati
- Valori correnti ⇒ Mantenuti
- Flag Globali ⇒ Mantenuti
- Variabili Globali ⇒ Mantenuti

• **EXPG (Esecuzione programma in parallelo)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	EXPG	Numero Programma	(Numero Programma)	CC

Avvia i programmi da quello definito nell'operando 1 a quello definito nell'operando 2, e li esegue in parallelo. L'operando 2 è opzionale. Il tentativo di avvio di uno o più programmi già in esecuzione genererà l'errore A57. Verificare la struttura del programma.

Esempio 1	EXPG	1	3	Esegue i programmi 1, 2 e 3.

- **ABPG (Termine programma in parallelo)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	ABPG	Numero Programma	(Numero Programma)	CC

Termina i programmi da quello definito nell'operando 1 a quello definito nell'operando 2, in esecuzione in parallelo. L'operando 2 è opzionale.

Esempio 1	ABPG	10	12	Termina i programmi 10, 11 e 12.

- **SSPG (Sospensione programma)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SSPG	Numero Programma	(Numero Programma)	CC

Sospende i programmi da quello definito nell'operando 1 a quello definito nell'operando 2, in esecuzione in parallelo. L'operando 2 è opzionale. La pausa interrompe anche l'eventuale comando in esecuzione nel programma.

Esempio 1	SSPG	10	12	Sospende i programmi 10, 11 e 12.

- **RSPG (Ripresa programma sospeso)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	RSPG	Numero Programma	(Numero Programma)	CC

Riprende i programmi sospesi con SSPG, a partire da quello definito nell'operando 1 fino a quello definito nell'operando 2. L'operando 2 è opzionale. Il sistema riprenderà l'esecuzione dal passo in cui era stato arrestato, continuando un'eventuale operazione interrotta.

Esempio 1	RSPG	10	12	Riprende i programmi 10, 11 e 12.

3.1.10 Gestione delle posizioni

- **PGET (Lettura posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PGET	Numero Asse	Numero Posizione	CC

Legge la quota relativa all'asse specificato nell'operando 1 dalla posizione specificata nell'operando 2, e la salva nella variabile 199. Se la posizione specificata non fosse definita (valore dell'asse non impostato), il comando PGET non verrebbe eseguito.

Esempio 1	PGET	2	3	Legge e salva nella variabile 199 il valore della posizione 3 sull'asse Y (asse 2).
Esempio 2	LET	1	2	Assegna 2 alla variabile 1.
	LET	2	3	Assegna 3 alla variabile 2.
	PGET	*1	*2	Legge e salva nella variabile 199 il valore della posizione 3 sull'asse Y (asse 2).

- **PPUT (Scrittura posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PPUT	Numero Asse	Numero Posizione	CP

Legge la quota desiderata dalla variabile 199, e la scrive nella posizione specificata nell'operando 2 relativa all'asse specificato nell'operando 1.

Esempio 1	LET	199	150	Assegna alla variabile 199 il valore 150.
	PPUT	2	3	Legge e salva il contenuto della variabile 199 (150) nella posizione 3 sull'asse Y (asse 2).
Esempio 2	LET	199	150	Assegna 199 alla variabile 150.
	LET	1	2	Assegna 2 alla variabile 1.
	LET	2	3	Assegna 3 alla variabile 2.
	PPUT	*1	*2	Legge e salva il contenuto della variabile 199 (150) nella posizione 3 sull'asse Y (asse 2).

- **PCLR (Cancella posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PCLR	Numero Posizione	Numero Posizione	CP

Cancella le posizioni a partire da quella specificata nell'operando 1, arrivando a quella specificata nell'operando 2. I campi posizione saranno rimossi e non impostati a 0.000.

Esempio 1	PCLR	10	20	Cancella le posizioni dalla 10 alla 20 comprese.
Esempio 2	LET	1	10	Assegna 10 alla variabile 1.
	LET	2	20	Assegna 20 alla variabile 2.
	PCLR	*1	*2	Cancella le posizioni dalla 10 alla 20 comprese.

- **PCPY (Copia posizioni)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PCPY	Numero Posizione	Numero Posizione	CP

Copia la posizione specificata nell'operando 2 nella posizione specificata nell'operando 1.

Esempio 1	PCPY	20	10	Copia la posizione 10 nella 20.
Esempio 2	LET	1	10	Assegna 10 alla variabile 1.
	LET	2	20	Assegna 20 alla variabile 2.
	PCLR	*1	*2	Copia la posizione 20 nella 10.

• **PRED (Legge la posizione corrente)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PRED	Schema assi	Numero Posizione	CP

Legge la quota corrente degli assi definiti nell'operando 1 e la salva nella posizione definita nell'operando 2.

Esempio 1	PRED	11	10	Legge la quota corrente in X ed in Y e ne salva i valori nella posizione 10.
Esempio 2	LET PRED	1 *1	3 10	Assegna 3 alla variabile 1. Legge la quota corrente in X ed in Y e ne salva i valori nella posizione 10. La combinazione degli assi può essere definita indirettamente; verrà tradotta in binario dal software.

• **PRDQ (Legge la posizione corrente)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PRDQ	Numero Asse	Numero Variabile	CP

Legge la quota corrente dell'asse definito nell'operando 1 e la salva nella variabile definita nell'operando 2.

Esempio 1	PRDQ	2	100	Legge la quota corrente in Y e ne salva il valore nella variabile 100.

• **PTST (Verifica dati posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PTST	Schema Assi	Numero Posizione	CC

Verifica se le quote della posizione specificata nell'operando 2, relative agli assi specificati nell'operando 1, siano valide. L'output verrà impostato ad OFF se i dati analizzati sono validi.

Esempio 1	PTST	11	10	300	Verifica la validità dei dati relativi agli assi X ed Y della posizione 10, ed imposta l'uscita 300 in base al risultato dell'analisi (ON dati invalidi, OFF dati validi).

- **PVEL (Imposta velocità posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PVEL	Velocità	Numero Posizione	CP

Assegna alla posizione impostata nell'operando 2 il valore di velocità definito nell'operando 1. La velocità deve essere espressa in mm/s.
(Nota) Se il valore impostato è negativo si otterrà un allarme ad ogni operazione di movimento diretta a quella posizione.

Esempio 1	PVEL	100	10	Assegna una velocità di 100 mm/s alla posizione 10.

- **PACC (Imposta accelerazione posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PACC	Accelerazione	Numero Posizione	CP

Assegna alla posizione impostata nell'operando 2 il valore di accelerazione definito nell'operando 1. L'accelerazione deve essere espressa in percentuale di G (accelerazione di gravità). Sono ammesse due cifre decimali.
(Nota) L'impostazione dell'accelerazione tramite questo comando non è soggetta a verifica di pertinenza dei dati introdotti.

Esempio 1	PACC	0.35	10	Assegna un'accelerazione di 0.35 G alla posizione 10.

• **PDCL (Imposta decelerazione posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PDCL	Decelerazione	Numero Posizione	CP

Assegna alla posizione impostata nell'operando 2 il valore di decelerazione definito nell'operando 1. La decelerazione deve essere espressa in percentuale di G (accelerazione di gravità). Sono ammesse due cifre decimali.
(Nota) L'impostazione della decelerazione tramite questo comando non è soggetta a verifica di pertinenza dei dati introdotti.

Esempio 1	PDCL	0.35	10	Assegna una decelerazione pari a 0.35 G alla posizione 10.

• **PAXS (Legge lo schema assi)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PAXS	Numero Variabile	Numero Posizione	CP

Memorizza nella variabile definita nell'operando 1 lo schema assi relativo alla posizione definita nell'operando 2.

Esempio 1	PAXS	1	98	Salva nella variabile 1 lo schema assi relativo alla posizione 98.

Il valore ottenuto nei quattro casi seguenti è rispettivamente:

No.(Name)	Axis1	Axis2	Axis3	Axis4	
98( )	200.000	100.000			0 0 1 1 = 2 + 1 = 3
99( )	350.000		120.000		0 1 0 1 = 4 + 1 = 5
100( )					0
101( )				180.000	1 0 0 0 = 8

- **PSIZ (Verifica dimensione dati posizione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PSIZ	Numero Variabile	Uso Proibito	CP

In base all'impostazione del parametro No. 23 degli *Other Parameters* cambia l'effetto dell'istruzione.

- Parametro No. 23 = 0 ⇒ Il controller salva nella variabile impostata nell'operando 1 il numero massimo di posizioni memorizzabili, a prescindere del numero di variabili già salvate.
- Parametro No. 23 = 1 ⇒ Il controller salva nella variabile impostata nell'operando 1 il numero di posizioni al momento memorizzate.

Esempio 1	PSIZ	1

- Parametro No. 23 = 0 ⇒ Il controller salva nella variabile 1 il numero massimo di posizioni memorizzabili, a prescindere del numero di variabili già salvate.
- Parametro No. 23 = 1 ⇒ Il controller salva nella variabile 1 il numero di posizioni al momento memorizzate.

- **GVEL (Memorizza velocità impostata)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GVEL	Numero Variabile	Numero Posizione	CP

Legge il valore di velocità impostato per la posizione definita nell'operando 2 e lo memorizza nella variabile definita nell'operando 1.

Esempio 1	GVEL	100	10	Memorizza nella variabile 100 la velocità impostata per la posizione 10.

- **GACC (Memorizza accelerazione impostata)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GACC	Numero Variabile	Numero Posizione	CP

Legge il valore di accelerazione impostato per la posizione definita nell'operando 2 e lo memorizza nella variabile definita nell'operando 1.

Esempio 1	GACC	100	10	Memorizza nella variabile 100 l'accelerazione impostata per la posizione 10.

• **GDCL (Memorizza decelerazione impostata)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GDCL	Numero Variabile	Numero Posizione	CP

Legge il valore di decelerazione impostato per la posizione definita nell'operando 2 e lo memorizza nella variabile definita nell'operando 1.

Esempio 1	GDCL	100	10	Memorizza nella variabile 100 la decelerazione impostata per la posizione 10.

3.1.11 Controllo degli attuatori

- **VEL (Imposta velocità)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	VEL	Velocità	Uso Proibito	CP

Imposta il valore di velocità definito nell'operando 1 per le operazioni CP. La velocità è espressa in mm/s.

Esempio 1	VEL	100	Imposta la velocità a 100 mm/s.

- **VELS (Esclusivo per SCARA: Imposta fattore di velocità)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	VELS	Fattore di Velocità	Uso Proibito	CP

Imposta il fattore di velocità come definito nell'operando 1 per le operazioni PTP (point-to-point). Il fattore di velocità è espresso in unità percentuali (50 %).

Esempio 1	VELS	50	Imposta al 50 % la velocità di esecuzione dei movimenti PTP.

- **OVRD (Override)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	OVRD	Fattore di Velocità	Uso Proibito	CP

Imposta il fattore di riduzione della velocità come definito nell'operando 1, sia per le operazioni CP che per le operazioni PTP (point-to-point).

Esempio 1	VEL	150	Imposta a 150 mm/s la velocità di esecuzione dei movimenti CP.
	VELS	50	Imposta al 50% la velocità di esecuzione dei movimenti PTP.
	OVRD	50	Imposta al 50% la velocità di esecuzione dei movimenti. I movimenti CP verranno quindi effettuati a 75 mm/s, mentre i movimenti PTP verranno effettuati al 25% della velocità.

• **ACC (Imposta accelerazione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	ACC	Accelerazione	Uso Proibito	CP

Imposta il valore di accelerazione definito nell'operando 1 per le operazioni CP. L'accelerazione è espressa in percentuale di G.

Esempio 1	ACC	0.3	Imposta l'accelerazione a 0.3 G.

• **ACCS (Esclusivo per SCARA: Imposta fattore di accelerazione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	ACCS	Fattore di Accelerazione	Uso Proibito	CP

Imposta il fattore di accelerazione come definito nell'operando 1 per le operazioni PTP (point-to-point). Il fattore di accelerazione è espresso in unità percentuali (50 %).

Esempio 1	ACCS	50	Imposta al 50 % del massimo ammesso l'accelerazione di esecuzione dei movimenti PTP.

- **DCL (Imposta decelerazione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DCL	Decelerazione	Uso Proibito	CP

Imposta il valore di decelerazione definito nell'operando 1 per le operazioni CP. La decelerazione è espressa in percentuale di G.

Esempio 1	DCL	0.3	Imposta la decelerazione a 0.3 G.

- **DCLS (Esclusivo per SCARA: Imposta fattore di decelerazione)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DCLS	Fattore di Decelerazione	Uso Proibito	CP

Imposta il fattore di decelerazione come definito nell'operando 1 per le operazioni PTP (point-to-point). Il fattore di decelerazione è espresso in unità percentuali (50 %).

Esempio 1	DCLS	50	Imposta al 50 % del massimo ammesso la decelerazione di esecuzione dei movimenti PTP.

- **VLMX (Esclusivo per movimenti lineari: Imposta velocità massima)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	VLMX	Uso Proibito	Uso Proibito	CP

Imposta la velocità dei movimenti successivi al valore VLMX di velocità (Normalmente il valore massimo). Il valore di velocità è definito nel parametro *Axis Specific Parameter* No. 29.

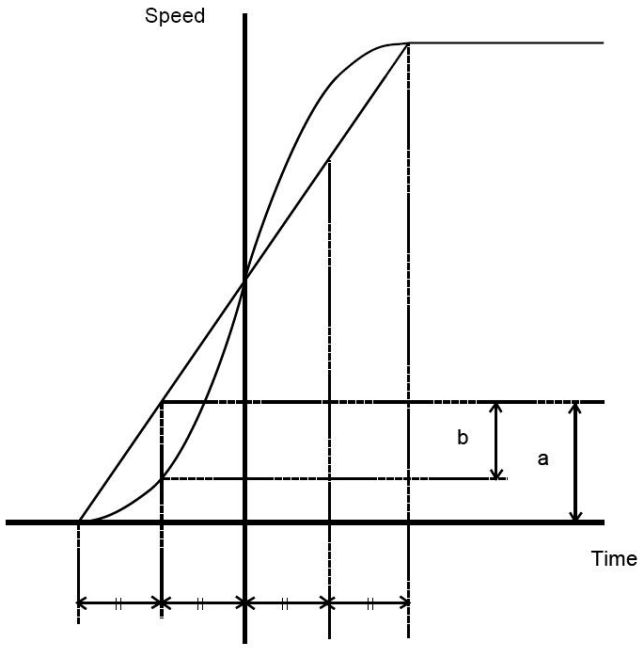
Esempio 1	VLMX		
	MOVP	5	Raggiunge la posizione 5 alla velocità VLMX.

- **SCRV (Imposta il fattore della sigmoide)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SCRV	Fattore	Uso Proibito	CP

Imposta il fattore del controllo sulla traiettoria sigmoidea dell'attuatore in fase di accelerazione. Se il fattore non è impostato o se lo si imposta a zero il profilo di moto sarà trapezoidale.

$$\frac{b}{a} \times 100(\%)$$



Esempio 1	SCRV	30	Imposta il fattore sigmoideo al 30%.

- **OFST (Imposta Offset)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	OFST	Schema Assi	Valore Offset	CP

Modifica le coordinate di moto sugli assi definiti nell'operando 1, aggiungendovi il valore di offset impostato nell'operando 2, quando vengono effettuati degli spostamenti. L'offset è impostato in mm, e può assumere valori negativi.
(Nota): Il comando di OFST è relativo al programma in cui viene impostato. Programmi in esecuzione in parallelo non godono dell'impostazione dell'OFST.

Esempio 1	OFST	110	50	Imposta un offset di 50 mm sugli assi Z e Y (3 e 2).
Esempio 2	LET	1	6	Assegna 6 alla variabile 1.
	OFST	*1	50	Imposta un offset di 50 mm sugli assi Z e Y (3 e 2).
Esempio 2	LET	1	30	Assegna 30 alla variabile 1.
	OFST	110	*1	Imposta un offset di 30 mm sugli assi Z e Y (3 e 2).

- **DEG (Imposta arco)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DEG	Angolo	Uso Proibito	CP

Imposta l'angolo di divisione per i moti interpolati concessi dal comando CIR o ARC. I punti di passaggio verranno quindi intervallati del valore scelto. L'angolo è assegnabile in un range tra 0 e 120 gradi. Impostando 0 il valore di divisione verrà scelto in automatico in base ai dati.
(Nota): Il controller ha un'impostazione di default dell'angolo di divisione, specificata nel parametro *All Axis Parameter* No. 30. Non è quindi necessario specificarlo con il comando DEG.

Esempio 1	DEG	10	Imposta l'angolo di divisione a 10 gradi.

- **BASE (Imposta l'asse di riferimento)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	BASE	Numero Asse	Uso Proibito	CP

Rinumera gli attuatori, partendo dall'asse definito nell'operando 1, impostandolo al valore di asse principale (numero 1).

Esempio 1	BASE	5	Il quinto asse verrà d'ora in poi considerato l'asse 1.
	HOME	1	L'asse 1 (asse 5 della precedente numerazione) effettua la ricerca origine.
	HOME	10	L'asse 2 (asse 6 della precedente numerazione) effettua la ricerca origine.

- **GRP (Imposta un gruppo di assi)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GRP	Schema Assi	Uso Proibito	CP

Autorizza la leggibilità dei valori nella tabella punti per i soli assi definiti dallo schema riportato nell'operando 1. Il programma valuterà come non impostati i valori degli altri assi.

Esempio 1	GRP	11		Considera validi solo i dati relativi agli assi X e Y.
	CIR2	1	2	Il comando non genererà errori di impostazione traiettoria, anche se i campi di Z e/o R dovessero contenere dati.

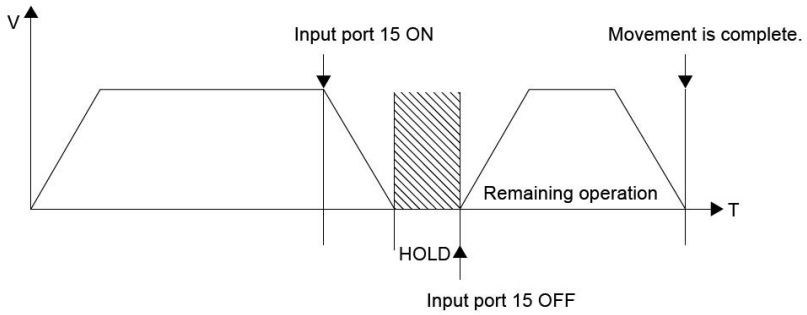
• **HOLD (Dichiara segnale di pausa)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	HOLD	Input, Flag	Metodologia	CP

Assegna ad un ingresso o ad un flag la valenza di segnale di pausa per tutte le operazioni di moto. Se il sistema sta già compiendo un moto, decelererà fino a fermarsi impostando ad ON l'ingresso o il flag definito nell'operando 1, per proseguire e completare il moto non appena l'ingresso stabilito tornerà allo stato OFF.

- Tipologia di Hold
- 0 = Contatto di tipo A (decelera fino a fermarsi)
- 1 = Contatto di tipo B (decelera fino a fermarsi)
- 2 = Contatto di tipo C (decelera fino a fermarsi, quindi spegne il servo motore)

Esempio 1	HOLD	15	0	Gli assi decelereranno fino a fermarsi quando l'ingresso 15 viene impostato ad ON.



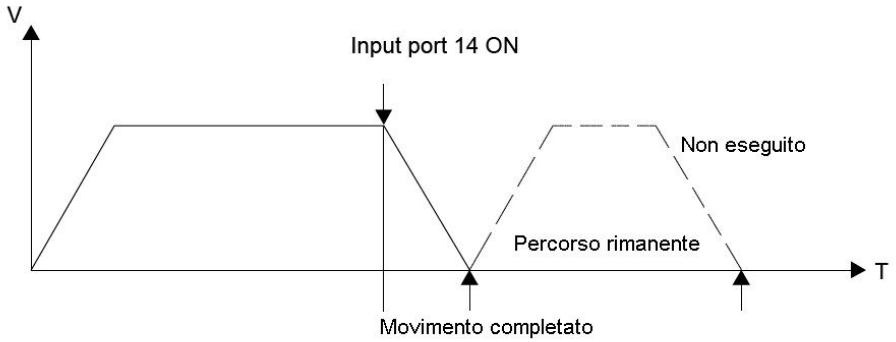
• **CANC (Dichiara segnale di abort)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	CANC	Input, flag	Metodologia	CP

Assegna ad un ingresso o ad un flag la valenza di segnale di abort per tutte le operazioni di moto. Se il sistema sta già compiendo un moto, decelererà fino a fermarsi impostando ad ON l'ingresso o il flag definito nell'operando 1.

- Tipologia di ABORT
- 0 = Contatto di tipo A (decelera fino a fermarsi)
- 1 = Contatto di tipo B (decelera fino a fermarsi)

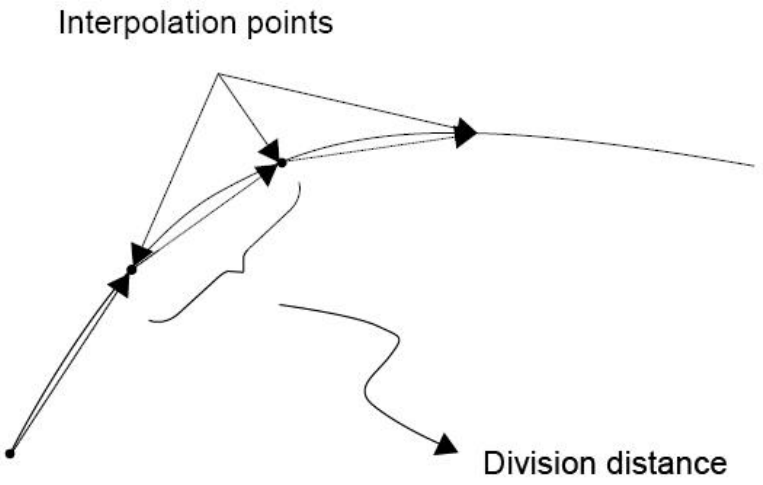
Esempio 1	CANC	14	0	Gli assi decelereranno fino a fermarsi impostando ad ON l'ingresso 14.



• **DIS (Imposta spline)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DIS	Distanza	Uso Proibito	CP

Imposta il segmento di divisione per i moti interpolati concessi dal comando PSPL. I punti di passaggio, usati per il calcolo dell'interpolazione, verranno quindi intervallati del valore scelto. Impostando 0 il valore di divisione verrà scelto in automatico in base ai dati. La distanza è espressa in mm.



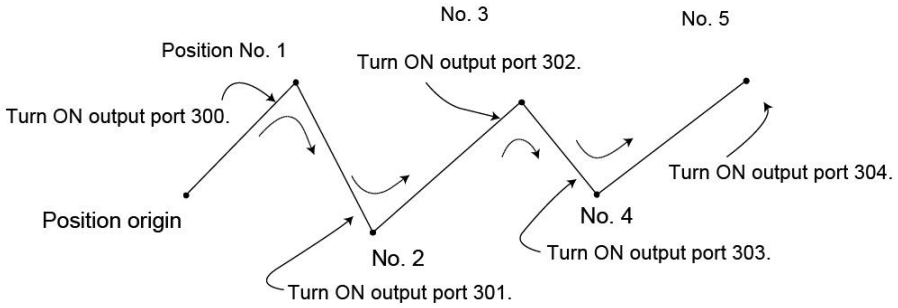
Esempio 1	DIS	10	Imposta la distanza di riferimento a 10 mm.

• **POTP (Imposta output PATH)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	POTP	0 o 1	Uso Proibito	CP

Imposta la modalità di funzionamento del campo Output del comando PATH e del comando PSPL, in base a quanto segue:

- POTP [Operando 1 = 0] ON al completamento del comando.
  L'uscita o il flag definito nel comando PSPL sarà impostato ad ON al completamento del comando.
- POTP [Operando 1 = 1] ON al completamento, incrementale ad ogni punto attraversato.
  L'uscita o il flag definito nel comando PSPL sarà impostato ad ON al completamento del comando. Durante il moto l'uscita definita verrà incrementata e impostata ad ON al passaggio da ogni punto intermedio.

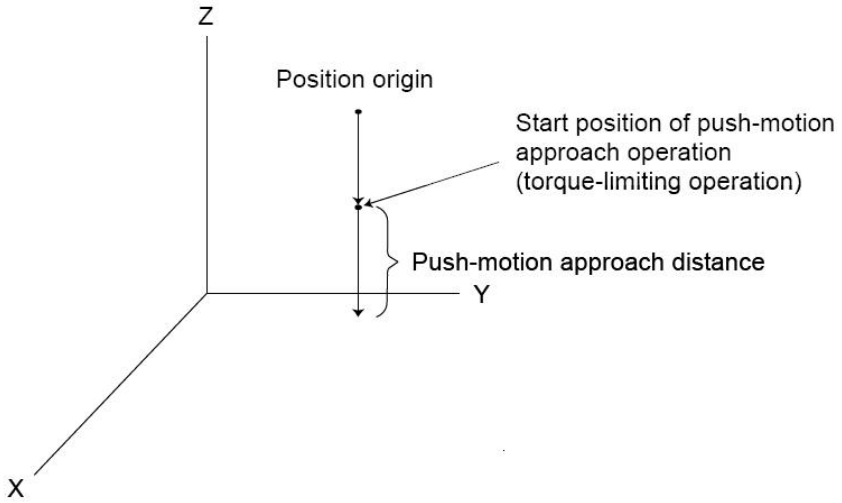


Esempio 1	POTP	1			
	PATH	1	5	300	Imposta sequenzialmente ad ON le uscite dalla 300 alla 304 ogni volta che si avvicina ad un punto di passaggio memorizzato.

- **PAPR (Imposta PUSH)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PAPR	Distanza	Velocità	CP

Definisce le operazioni da compiere nell'esecuzione di un comando di PUSH. Imposta nell'operando 1 lo spazio da percorrere effettuando il controllo di coppia, mentre nell'operando 2 definisce la velocità con cui procedere in PUSH.

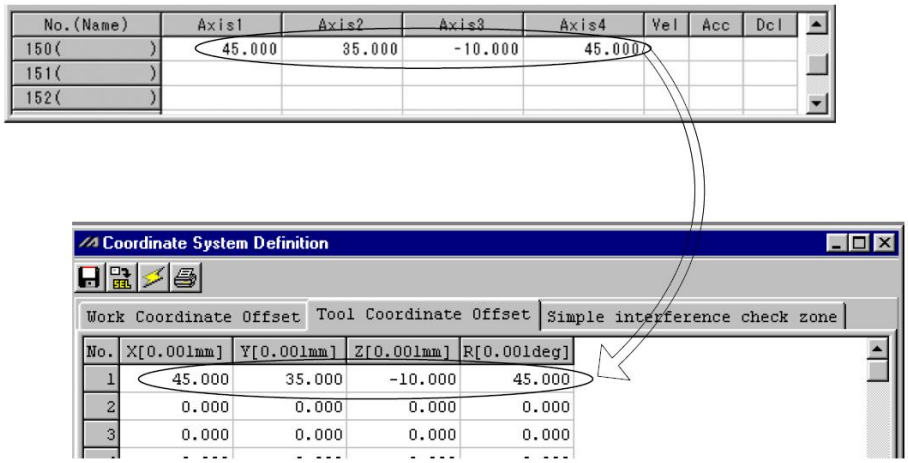


Esempio 1	PAPR	100	30	Definisce l'operazione di PUSH, da compiere a 30 mm/s avanzando per 100 mm.
	MOVP	10		
	PUSH	11		Compie il moto di PUSH, come impostato in precedenza.

- **DFTL (Esclusivo per SCARA: definisce Tool)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DFTL	Numero Tool	Numero Posizione	CP

Legge i dati relativi alla posizione definita nell'operando 2, e li utilizza per impostare i valori di offset del tool definito nell'operando 1. Campi invalidi nella posizione verranno inizializzati come offset nulli nel Tool.



Esempio 1	DFTL	1	150	

- **STLT (Esclusivo per SCARA: Sceglie Tool)**

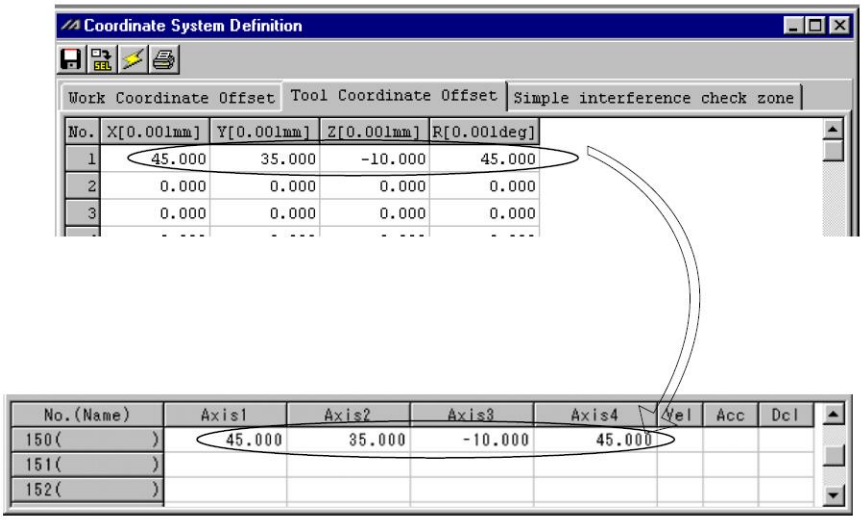
Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	STLT	Numero Tool	Uso Proibito	CP

Imposta il sistema di riferimento dell'utensile in base al numero specificato nell'operando 1.

- **GTTL (Esclusivo per SCARA: Importa Tool)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GTTL	Numero Tool	Numero Posizione	CP

Legge i dati relativi al tool definito nell'operando 1, e li memorizza nella posizione definita nell'operando 2.

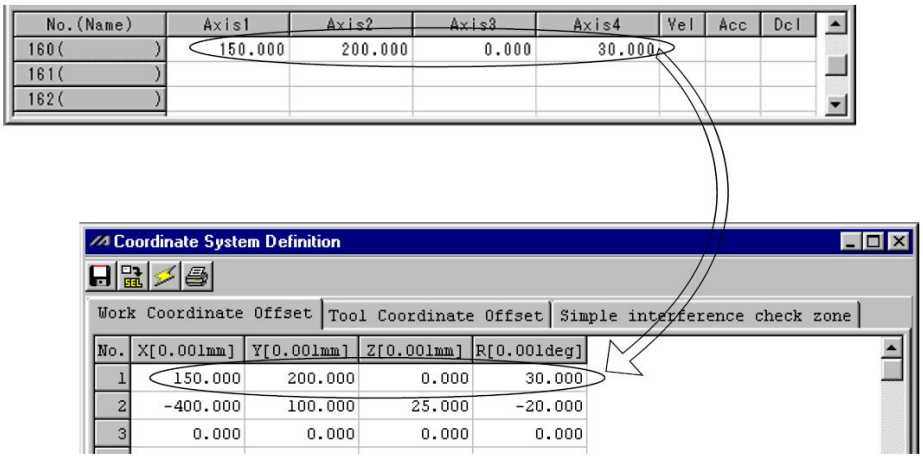


Esempio 1	GTTL	1	150

- **DFWK (Esclusivo per SCARA: definisce sistema di coordinate Work)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DFWK	Numero Work	Numero Posizione	CP

Legge i dati relativi alla posizione definita nell'operando 2, e li utilizza per impostare i valori di offset del sistema di coordinate di lavoro definito nell'operando 1. Campi invalidi nella posizione verranno inizializzati come offset nulli nel Work.



Esempio 1	DFWK	1	160

- **STWK (Esclusivo per SCARA: Sceglie Work)**

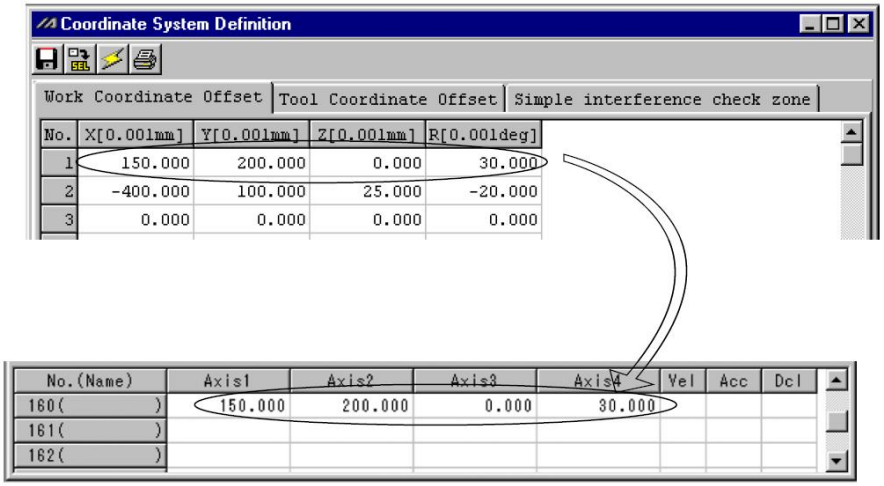
Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	STWK	Numero Work	Uso Proibito	CP

Imposta il sistema di riferimento di lavoro in base al numero specificato nell'operando 1.

- **GTWK (Esclusivo per SCARA: Importa Work)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GTWK	Numero Work	Numero Posizione	CP

Legge i dati del sistema di riferimento di lavoro definito nell'operando 1, e li memorizza nella posizione definita nell'operando 2.


Coordinate System Definition
Work Coordinate Offset
Tool Coordinate Offset
Simple interference check zone
No. X[0.001mm] Y[0.001mm] Z[0.001mm] R[0.001deg]
1 150.000 200.000 0.000 30.000
2 -400.000 100.000 25.000 -20.000
3 0.000 0.000 0.000 0.000
No.(Name) Axis1 Axis2 Axis3 Axis4 Vel Acc Dcl
160( ) 150.000 200.000 0.000 30.000
161( )
162( )

Esempio 1	GTWK	1	160	

• **RIGH (Esclusivo per SCARA: Imposta a destro il gomito)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	RIGH	Uso Proibito	Uso Proibito	PE

Cambia l'orientamento attuale del robot, impostandolo a gomito destro. Se il robot al momento è ad orientamento sinistro, il secondo giusto si allineerà al primo, formando una linea retta. Se il robot è già ad orientamento destro, non succederà nulla.

• **LEFT (Esclusivo per SCARA: Imposta a sinistro il gomito)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	LEFT	Uso Proibito	Uso Proibito	PE

Cambia l'orientamento attuale del robot, impostandolo a gomito sinistro. Se il robot al momento è ad orientamento destro, il secondo giusto si allineerà al primo, formando una linea retta. Se il robot è già ad orientamento sinistro, non succederà nulla.

• **PTPR (Esclusivo per SCARA: Vincola l'orientamento destro)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PTPR	Uso Proibito	Uso Proibito	PE

Imposta l'orientamento destro come riferimento per i moti PTP. Tutte le operazioni moto point to point saranno quindi vincolate al gomito destro. Scegliendo una posizione non raggiungibile se non con orientamento sinistro, il controller andrà in errore.

• **PTPL (Esclusivo per SCARA: Vincola l'orientamento sinistro)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PTPL	Uso Proibito	Uso Proibito	PE

Imposta l'orientamento sinistro come riferimento per i moti PTP. Tutte le operazioni moto point to point saranno quindi vincolate al gomito sinistro. Scegliendo una posizione non raggiungibile se non con orientamento destro, il controller andrà in errore.

• **PTPD (Esclusivo per SCARA: Vincola l'orientamento corrente)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PTPD	Uso Proibito	Uso Proibito	PE

Imposta l'orientamento corrente come riferimento per i moti PTP. Tutte le operazioni moto point to point saranno quindi vincolate al gomito riscontrato all'esecuzione di questa istruzione. Scegliendo una posizione non raggiungibile se non con orientamento opposto, il controller andrà in errore.

• **PTPE (Esclusivo per SCARA: Svincola l'orientamento)**

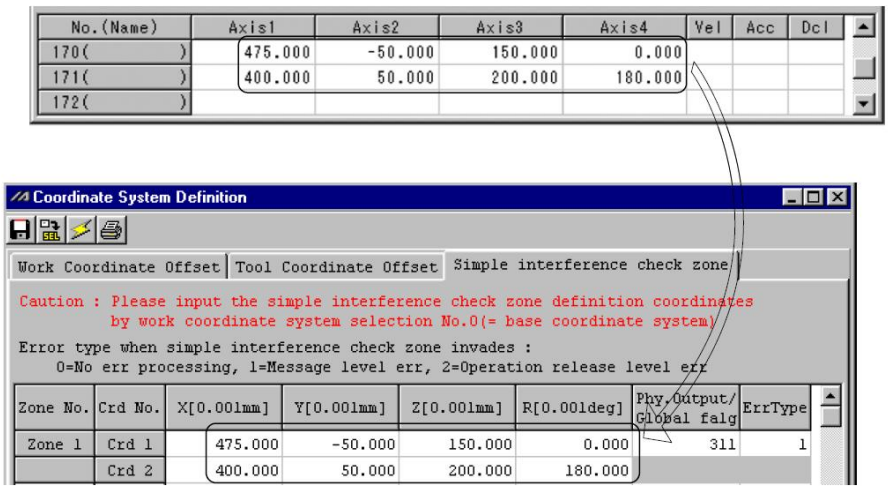
Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PTPE	Uso Proibito	Uso Proibito	PE

Imposta l'orientamento attuale come preferenza per i moti PTP. Tutte le operazioni moto point to point saranno quindi impostate con l'orientamento corrente. Qualora la posizione si dimostri non raggiungibile, il controller provvederà al cambio di orientamento. vincolate al gomito sinistro. Scegliendo una posizione non raggiungibile, nè con orientamento destro nè con il sinistro, il controller andrà in errore.

- **DFIF (Esclusivo per SCARA: Definisce spazio di interferenza)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	DFIF	Numero Zona	Numero Posizione	CP

Imposta le due posizioni consecutive, a partire da quella definita nell'operando 2, come zona di interferenza semplice definita nell'operando 1. La posizione definita nell'operando 2 verrà utilizzata per definire il primo estremo della zona di interferenza, mentre la posizione successiva verrà utilizzata per definire il secondo estremo.

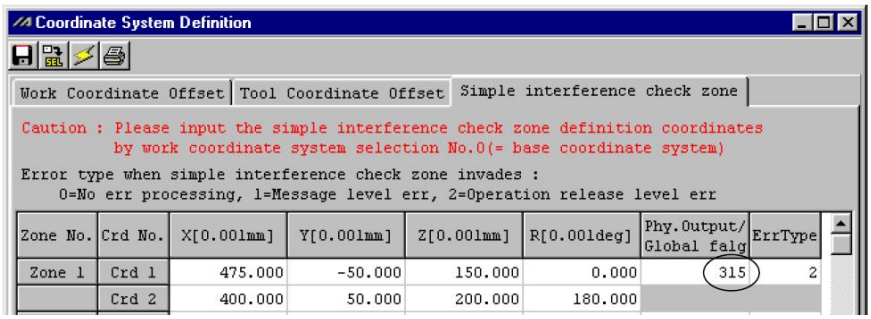


Esempio 1	DFIF	1	170	

- **SOIF (Esclusivo per SCARA: Specifica uscita spazio di interferenza)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SOIF	Numero Zona	Numero Uscita, Flag	CP

Imposta l'uscita definita nell'operando 2, come riferimento di sistema per la zona di interferenza semplice definita nell'operando 1. Appena lo SCARA si introduce in questa zona, l'uscita verrà impostata ad ON.



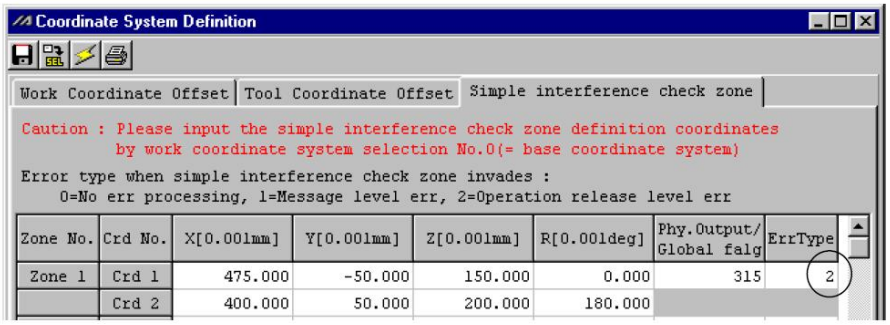
Esempio 1	SOIF	1	315

- **SEIF (Esclusivo per SCARA: Specifica errore spazio di interferenza)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SEIF	Numero Zona	0 , 1 o 2	CP

Imposta la metodologia di errore legata al passaggio nella zona di interferenza specificata nell'operando 1.

- 0: Nessun Errore
- 1: Messaggio di errore
- 2: Cancellazione delle operazioni correnti



Esempio 1 SEIF 1 2

- **GTIF (Esclusivo per SCARA: Importa zona interferenza)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	GTIF	Numero Zona	Numero Posizione	CP

Legge i valori della zona di interferenza definita nell'operando 1 e li salva nelle due posizioni consecutive a partire da quella specificata nell'operando 2.

• **HOME (Esclusivo per assi: Ricerca origine)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	HOME	Schema Assi	Uso Proibito	PE

Effettua la sequenza di ricerca origine sugli assi specificati nell'operando 1. Il servomotore di ciascun asse interessato verrà automaticamente acceso per eseguire il comando. La variabile di Output verrà impostata ad OFF durante l'esecuzione del comando, e impostata al valore ON al completamento della sequenza di ricerca origine.

Esempio 1	HOME	1100	Effettua la ricerca origine per gli assi 3 e 4.

3.1.12 Movimentazione degli attuatori

- **SVΞΞ (Gestione servomotori)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	SVΞΞ	Schema Assi	Uso Proibito	PE

Accende/spegne il servomotere degli assi specificati nell'operando 1.
(Nota) Lavorando con uno SCARA i servomotori di tutti gli assi verranno accesi o spenti, a prescindere dallo schema assi impostato.

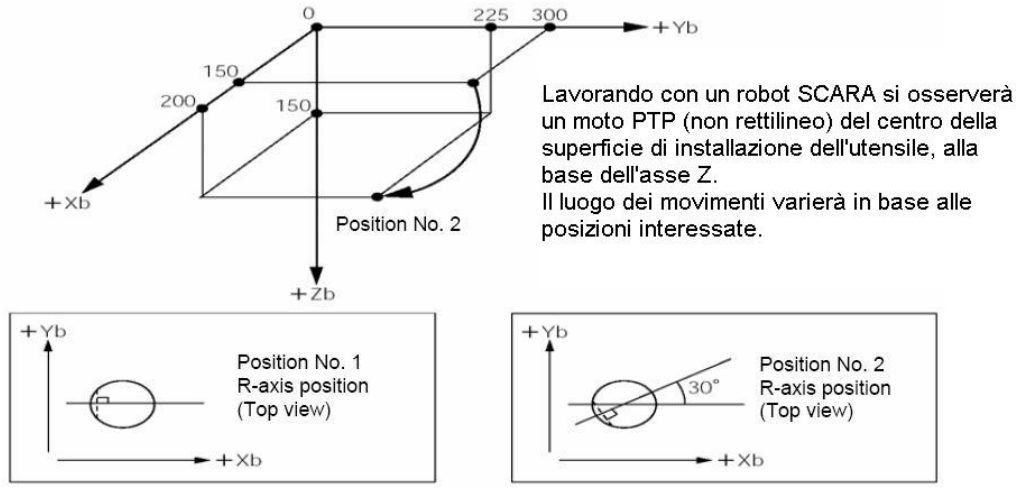
SVΞΞ
- ↘ ON ... Accende il servomotore
- ↘ OFF ... Spegne il servomotore

Esempio 1	SVON	1100	Accende il servomotore degli assi 3 e 4.
Esempio 2	SVOF	1100	Spegne il servomotore degli assi 3 e 4.

- **MOVP (Movimento Point to point)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MOVP	Numero Posizione	Uso Proibito	PE

Muove l'attuatore fino alla posizione definita nell'operando 1. La variabile di Output verrà impostata ad OFF durante l'esecuzione del comando, e impostata al valore ON al raggiungimento della posizione desiderata.

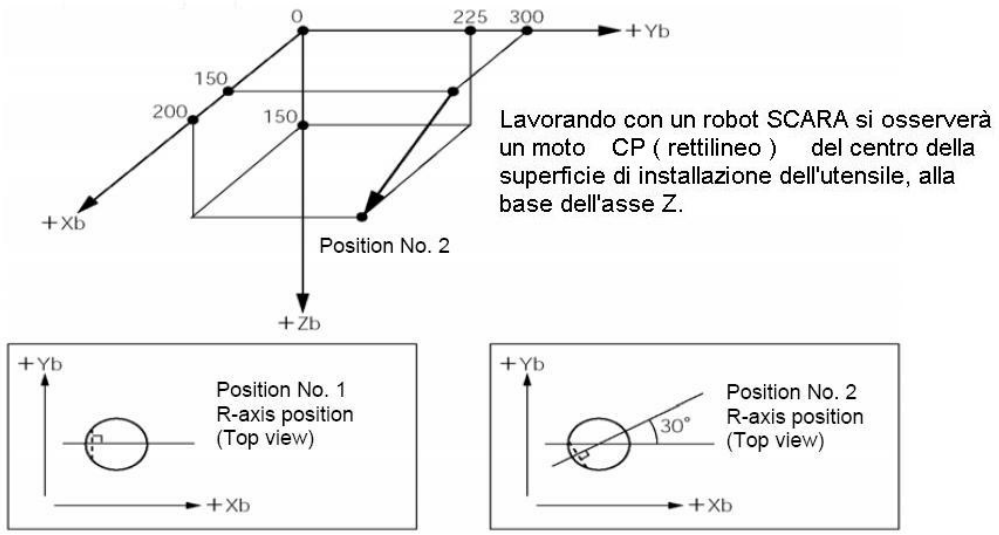


Esempio 1	MOVP	11	Muove gli assi per raggiungere le quote specificate nella posizione 11.

- **MOVL (Movimento Interpolato lineare)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MOVL	Numero Posizione	Uso Proibito	PE

Muove l'attuatore fino alla posizione definita nell'operando 1, con moto interpolato lineare. La variabile di Output verrà impostata ad OFF durante l'esecuzione del comando, e impostata al valore ON al raggiungimento della posizione desiderata.



Esempio 1	MOVL	2	Muove gli assi per raggiungere le quote specificate nella posizione 2, con interpolazione.

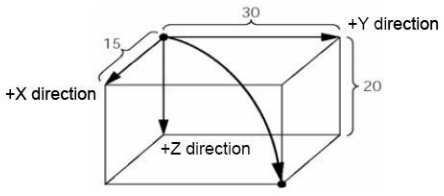
• **MVPI (Movimento Incrementale PTP)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MVPI	Numero Posizione	Uso Proibito	PE

Muove l'attuatore con moto PTP non interpolato dalla posizione corrente per lo spazio impostato nella posizione definita nell'operando 1. La variabile di Output verrà impostata ad OFF durante l'esecuzione del comando, e impostata al valore ON al raggiungimento della posizione desiderata.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
5( )	200.000	150.000	50.000	45.000			
6( )	15.000	30.000	20.000	30.000			
7( )							

Percorso incrementale dalla posizione 5, dei valori specificati nella posizione 6.



Lavorando con un robot SCARA si osserverà un moto PTP (non rettilineo) del centro della superficie di installazione dell'utensile, alla base dell'asse Z.
Il luogo dei movimenti varierà in base alle posizioni interessate.

Esempio 1	MVPI	6	Muove gli assi per incrementare la posizione corrente delle quote specificate nella posizione 6, con moto PTP.

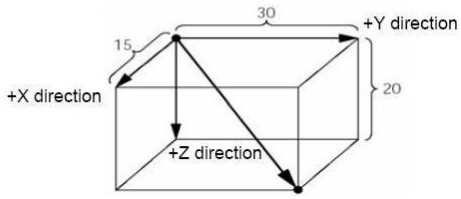
- **MVLI (Movimento Incrementale interpolato CP)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	MVLI	Numero Posizione	Uso Proibito	PE

Muove l'attuatore, con moto interpolato lineare, dalla posizione corrente per lo spazio impostato nella posizione definita nell'operando 1. La variabile di Output verrà impostata ad OFF durante l'esecuzione del comando, e impostata al valore ON al raggiungimento della posizione desiderata.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
5( )	200.000	150.000	50.000	45.000			
6( )	15.000	30.000	20.000	30.000			
7( )							

Percorso incrementale dalla posizione 5, dei valori specificati nella posizione 6.



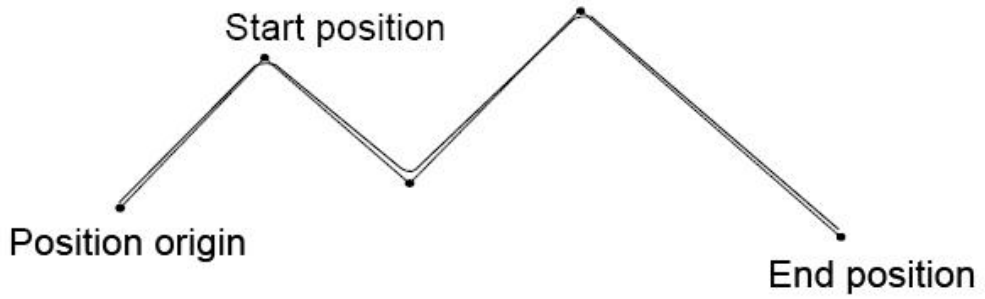
Lavorando con un robot SCARA si osserverà un moto CP (rettilineo) del centro della superficie di installazione dell'utensile, alla base dell'asse Z.

Esempio 1	MVLI	6	Muove gli assi per incrementare la posizione corrente delle quote specificate nella posizione 6, con moto interpolato lineare.

- **PATH (Movimento lungo un percorso)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	PATH	Posizione Iniziale	Posizione Finale	PE

Muove l'attuatore, con moto interpolato lineare, compiendo un percorso che copra le posizioni dalla posizione specificata nell'operando 1 alla posizione specificata nell'operando 2. La variabile di Output è gestita dal comando POTP (3.1.11).

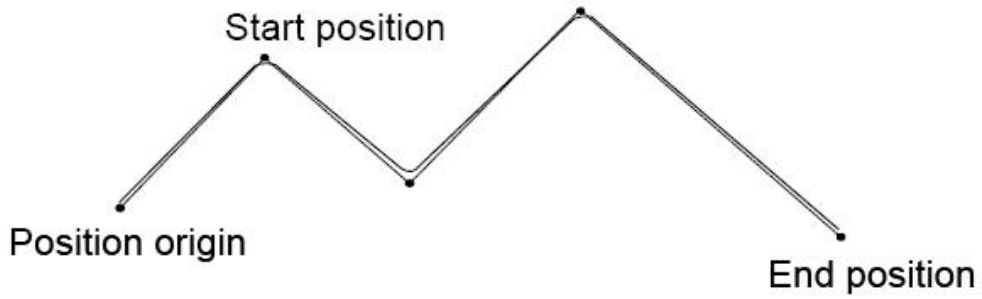


Esempio 1	MVLI	6	Muove gli assi per incrementare la posizione corrente delle quote specificate nella posizione 6, con moto interpolato lineare.

- **COMANDO (Breve def comando)**

Condizione (AND,OR)	Input	Comando	Operando 1	Operando 2	Output
Optional	Optional	COMANDO	Op 1	Op 2	Out

Spiegazione del comando...



Esempio 1	COMANDO	Op 1	Op 2	Output	Commento del comando.

3.1.13 IF strutturato

3.1.14 DO strutturato

3.1.15 Multi-Branch

3.1.16 Acquisizione informazioni sistema

3.1.17 Zone

3.1.18 Comunicazione

3.1.19 Operazioni sulle stringhe

3.1.20 Gestione pallet

3.1.21 Impostazione pallet

3.1.22 Movimenti su pallet

3.1.23 Pseudo-Ladder

3.1.24 Comandi estesi

SINTA S.r.l.
Via Sant'Uguzzone, 5
20126 - Milano, Italia
Tel: +39-02-27007238 - **Fax**: +39-02-27007858
E-Mail: info@sinta.it - **Internet**: http://www.sinta.it